



Escola de Camins
Escola Tècnica Superior d'Enginyeria de Camins, Canals i Ports
UPC BARCELONATECH

Posibilidad de la realidad virtual en entornos BIM

Treball realitzat per:

Soraya Jeannette Araujo Criollo

Dirigit per:

Javier Mora

Ignacio Valero

Máster en:

Ingeniería Estructural y de la Construcción

Barcelona, **febrero 2017**

Departament de construcció

TREBALL FINAL DE MÀSTER

Máster

Ingeniería Estructural y de la Construcción

Título

Posibilidad de la realidad virtual entorno BIM

Autor

Soraya Jeannette Araujo Criollo

Tutor

**Ignacio Valero
Javier Mora**

Especialidad

Construcción

Departamento

Construcción

Fecha

Febrero 2017

Agradecimientos

Primero y como más importante, me gustaría agradecer sinceramente a mis tutores de Tesis, Dr. Javier Mora e Ignacio Valero, por su esfuerzo, dedicación, conocimientos, sus orientaciones, su paciencia y sobre todo la motivación que han puesto en mí y en mi proyecto, han sido fundamentales para mi formación y ha sido también mi motor de motivación para desarrollar el presente documento.

Ellos me han hecho descubrir además la relación que guarda un mundo tan tecnológico como es el sector de los videojuegos con el sector de la construcción. A través de las reuniones que desde junio hasta el día de hoy hemos tenido semana a semana, donde gracias a ello, ellos han inculcado en mí un sentido de seriedad, responsabilidad y rigor, ya que sin su ayuda mi formación académica no se podría haber completado.

Agradezco al centro CIMNE¹ por brindarme la oportunidad de desarrollar mi proyecto en su sede, a los ingenieros Carles y Jordi de VITXU² por permitirme conocer su trabajo de primera mano y enseñarme las herramientas que ellos utilizan, al Centro de Realidad Virtual de Barcelona³ por la demostración realizada durante la visita a sus instalaciones, en particular para experimentar los sistemas CAVE⁴ y StereoWall.

También quiero agradecer a Arnau y Óscar, mis compañeros de batalla a lo largo de estos meses, todo su apoyo en este tiempo en que hemos compartido la misma línea de trabajo, con diferentes casos, dificultades de implementación y todos aquellos otros problemas que nos han ido surgiendo a lo largo de la elaboración de nuestras tesinas.

Por último, quiero agradecer a mi familia, en especial a mis padres y a mi hermano. Ellos son el motor de mi vida. Gracias a su infinita paciencia y a sus muchos consejos he podido lograr mis objetivos, mantener mis motivaciones y el esfuerzo diario que me han llevado al día de hoy a exponer mi tesis final de master.

¹ Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE), <http://www.cimne.com/>

² VIXTU Technologies, <http://www.vixtu.com/>

³ The Modeling, Visualization, Interaction and Virtual Reality Research Group (MOVING), Research Center for Visualization, Virtual Reality and Graphics Interaction (ViRVIG), <http://moving.cs.upc.edu/>

⁴ “Cave Automatic Virtual Environment”, también encontrado como “Computer Assisted Virtual Environment”

Motivación

El sector de la construcción a lo largo de los años ha experimentado cambios sustanciales en su manera de trabajar. En los años 80 todo se basaba en planos hechos en papel con plumillas donde la precisión del ingeniero, arquitecto y el delineante eran impecables, pero con el inconveniente del tiempo, ya que su puesta a punto llevaba detrás mucho trabajo. Por ejemplo, en la preparación de los planos se necesitaba a un equipo de varias personas, puesto que, más allá del diseño conceptual, era preciso generar las diferentes vistas (planta, alzado, perfil, perspectivas) que permitieran disponer de todos los detalles de la construcción. Mientras pequeños errores podían resolverse mediante raspado sobre los planos en papel vegetal, otras modificaciones en el diseño implicaban, a menudo, tener que repetir planos enteros. A partir de los años 90 empieza a coger fuerza una nueva alternativa, una novedosa forma de trabajo basada en la creciente utilización de ordenadores personales, el CAD (*Computer Aided Design*), que permitía hacer múltiples modificaciones sin necesidad de rehacer todos los dibujos.

Una manera de trabajar más rápida, sencilla y económica, aunque con el inconveniente de requerir una formación especializada: pocos profesionales sabían manejar este tipo de programas, a menudo ni siquiera sabían trabajar con los ordenadores donde se ejecutaban. Con el tiempo y la evolución en cuanto formas de trabajo, maneras de ejecutarlo y tecnología, han nacido nuevas herramientas y formas de utilizarlas. En la actualidad, en los países industrializados, el diseño y modelado con planos tal y como se ejecutaba antaño no es muy común. Las constructoras han optado por modelos digitalizados, ya que de esta manera se puede tener toda información bien documentada, ordenada y guardado en una carpeta de cualquier ordenador o, ahora más recientemente, en la nube.

Con esta filosofía de trabajo nace un nuevo concepto: la modelación digital en sentido amplio, bajo el paraguas de lo que se conoce por BIM (por sus siglas en inglés, *Building Information Modeling*, o Modelado de información de construcción). El objetivo de esta nueva y novedosa filosofía es, grosso modo, reducir los costes, mejorar la eficiencia y eficacia tanto en el trabajo de grupo como individual, a través de su conexión en tiempo real entre distintos usuarios que trabajan en el mismo modelo conectados a un archivo central común.

Además, es importante señalar que los procedimientos y herramientas BIM constituyen ya una metodología de obligado cumplimiento en determinados países, y que, más allá de tratarse de una competición entre programas, se trata de una filosofía de funcionamiento integradora y de amplia interoperabilidad, como veremos más adelante.

Con la fuerte evolución a nivel tecnológico a la que estamos sometidos en la actualidad, el sector de la Arquitectura, Construcción e Ingeniería (ACE), que tradicionalmente evoluciona en sus procedimientos y herramientas con una gran inercia, tiene una oportunidad para incorporar grandes innovaciones, primero mediante BIM y, segundo, mediante los avances en las tecnologías de la información y comunicaciones.

En particular, el concepto general es conectar de alguna forma el modelado BIM con la realidad virtual y/o aumentada (RV/RA), partiendo de la premisa de que ambas disciplinas conforman el trabajo de modelos 3D como elemento de trabajo esencial. El trabajo a desarrollar implica, en primer lugar, revisar bien que es RV/RA, explorar cómo podría trabajar o conectarse con la metodología BIM y, finalmente, proponer algunos casos de uso que sirvan para conocer el detalle de la tecnología y validar su utilidad en la construcción.

Finalmente, como ingeniera civil siempre me he planteado buscar nuevas formas de trabajo que nos permitan ser más eficientes, económicos y productivos. El foco principal de este proyecto sigue esa búsqueda de innovación: a partir de una idea tan simple en concepto y a la vez tan compleja en su ejecución, vincular la RV/RA – BIM y compararla con las formas de trabajo habituales, para valorar si pueden proporcionarnos medios para abordar proyectos más complejos, más sólidos o para, simplemente, ser más eficientes.

Abstracto

Para hablar de realidad virtual, aumentada y buscar la manera de cómo conectar con la metodología BIM, primero hay que definir qué es cada elemento, cómo se utilizan y en qué sectores. De forma simplificada, la realidad virtual es la recreación de un escenario mediante herramientas informáticas para generarnos la ilusión de encontrarnos en un entorno diferente al real. La primera idea que nos viene a la cabeza cuando se habla de RV es asociarla al sector del entretenimiento, como son los videojuegos, para situarnos en épocas pretéritas o futuristas donde luchamos o perseguimos los objetivos del juego. Sin embargo, el sector inmobiliario también lleva ya cierto tiempo trabajando con este concepto a la hora de atraer a potenciales compradores. Muestran cómo quedaría la obra mediante un recorrido virtual antes de que ésta se haya completado, a modo de canal comercial. Otros ámbitos donde también se utiliza la RV es en museos, promociones turísticas, en formación o en el campo médico, donde ya llevan un largo recorrido.

Por otro lado, la realidad aumentada, aunque en muchas ocasiones se llega a confundir con la realidad virtual, quizás debido a que utilizan tecnologías similares, presenta una diferencia de concepto importante. La realidad aumentada se centra en la visualización de información digital en un entorno físico del mundo real, es decir, son elementos tangibles que se asocian a su vez con elementos virtuales, creando una realidad física, palpable y a tiempo real en sitios reales. Se podría decir que todos o la gran mayoría de personas a diario ya utilizan este tipo de realidad ya que, si se quiere encontrar cómo llegar a un lugar determinado y no se sabe cómo, la gran mayoría de estos usuarios optan por buscar a través de alguna aplicación y no coger el típico callejero. Es decir, recurren a aplicaciones como Yelp o también el mapa de Google, donde desde el punto de vista conceptual se podría considerar como un ejemplo de realidad aumentada.

Respecto al BIM, dependiendo del autor podemos encontrar una definición u otra, donde la idea común que presentan podría sintetizarse como: la creación de procesos de generación y gestión de datos del proyecto constructivo durante su ciclo de vida, utilizando software dinámico de modelado en tres dimensiones a tiempo real, vinculado con otra información digital como son los flujos de trabajo, con el objetivo de reducir la pérdida de tiempo, recursos en el diseño y en la construcción.

Para poder implementar el modelado de la información (BIM) es necesario tener un software compatible. En el mercado actual hay una gran variedad de programas, pero si nos centramos en nuestro campo de estudio vamos a trabajar con los paquetes de Autodesk, por un lado, Revit y AutoCAD, por otro 3DS Studio MAX (3ds Max), que no está vinculado a entornos BIM, pero es ampliamente usado por los arquitectos para la representación fotorrealista de los edificios, lo que les aproxima a las herramientas RV/RA.

Si empezamos a explorar el uso que posee cada uno de los programas anteriores dentro de nuestro campo de estudio vemos, por ejemplo, que en Revit vamos a poder crear una tabla de planificación, conocer el tipo de material que se va a colocar o crear cualquier tipo de estructura, tanto en alzado como en tres dimensiones, incluso se puede modificar una parte de la misma como por ejemplo un pilar intermedio y automáticamente se modifica en todo el proyecto incluidos planos y vistas.

AutoCAD es una de las herramientas más longevas y utilizadas dentro del sector, posee una gran variedad de usos, permite crear, diseñar y modelizar planos tanto en 2D como 3D, pero en nuestro caso nos servirá sobre todo para la visualización de planos, comparación de instalaciones, cotas y detalles constructivos.

3ds Max tiene una implementación basada fundamentalmente en la visualización, diseño y recorrido en 3D, es una potente máquina para generar un renderizado⁵. Su uso habitual está ligado al sector de la producción y animación. Posee una carpeta de materiales y texturas que nos será útil más adelante para hacer la experiencia de RV más inmersiva y realista.

Unity 3d ha sido la herramienta escogida para implementar la realidad virtual. Puede definirse como un motor de videojuegos multiplataforma con contenidos interactivos en 3D, lo que nos permitirá diseñar recorridos virtuales por la edificación.

A nivel de implementación, hemos partido de dos modelos ya existentes del edificio B0 del Campus Nord, uno desarrollado en 3ds Max, que representa el diseño conceptual del módulo y que nos servirá para el estudio arquitectónico y estructural, y otro en formato Revit, preparado como modelo As-Built, donde analizaremos sus instalaciones y fontanería, y también nos servirá para la comparación de modelos.

Cada uno de estas dos representaciones del edificio nos sirven para estudiar dos casos de uso particulares.

En el primero de ellos exploraremos las posibilidades que ofrece el recorrido virtual para la construcción, tales como revisión de la arquitectura, etiquetado virtual de espacios, comparativa de planos, efectos de iluminación y disposición de mobiliario, etc.

El segundo caso de estudio se centra en la revisión de las instalaciones, como medio para evitar solapamientos entre diferentes proveedores, remates de la obra, etc.

⁵ Se entiende por *renderizado* o *render* la imagen resultante del procesado de la iluminación sobre un modelo 3D que incluye información de materiales y texturas, lo que proporciona una apariencia fotorealista.

Índice

Contenido

Agradecimientos	4
Motivación	5
Abstracto.....	7
Índice	9
1. Capítulo 1: Planteamiento	11
1.1. Introducción	11
1.2. Planteamiento del problema.....	11
1.3. Objetivos	11
1.3.1. Objetivos generales	11
1.3.2. Objetivos Específicos.....	12
1.4. Metodología	12
2. Capítulo 2: Estado de la tecnología	14
2.1. BIM	15
2.1.1. Ciclo de vida de un proyecto en BIM	16
2.1.3. Clasificación de dimensionamiento	18
2.1.4. Evolución de cambio de CAD a BIM	19
2.1.5. Niveles de desarrollo según el marco europeo.....	20
2.1.6. Formato universal IFC	21
2.2. Revit	21
2.3. Unity 5.....	22
2.4. Realidad virtual.....	23
2.5. Realidad aumentada	25
2.6. 3ds Studio Max.....	26
2.7. Realidad aumentada – BIM	27
3. Capítulo 3: Caso de estudio	28
4.1. Caso es estudio 1: Estructura + Arquitectura	33
4.2. Modelo gráfico 2: Instalaciones	48
4. Capítulo 4: Análisis del proyecto.....	58
4.1. Introducción	58
4.2. Desarrollo	58
4.3. Valoración y resultados	66
5. Capítulo 5: Conclusiones.....	67

6. Capítulo 6: Futuras líneas de trabajo	69
7. Referencias	70
Índice de figuras.....	72
Glosario	75
ANEXOS	76
A Anexo 1. Metodología BIM.....	76
B Anexo 2. Diseño del edificio en 3ds Max.....	76
C Anexo 3. Lenguaje de programación. Códigos en C Sharp.....	76

1. Capítulo 1: Planteamiento

1.1. Introducción

Tal y como se ha expresado en la motivación, este trabajo tiene como objetivo **explorar las posibilidades de la realidad virtual en el sector de la construcción**, vinculado con el uso de herramientas y metodologías BIM.

Para avanzar en este objetivo vamos a distinguir dos modelos de trabajo. El primero, a nivel teórico, que implica la revisión de las definiciones y conceptos para entender de manera más precisa los componentes en torno a la realidad virtual y el modelado BIM, así como sus posibilidades de aplicación. El segundo nivel plantea la manera de implementar la tecnología mediante dos casos de uso, lo que nos permite profundizar en el detalle de los aspectos prácticos y evaluar su alcance en el contexto de la obra.

Para ello se han empleado dos modelos distintos del edificio, el primero está asociado a la ejecución previa, la etapa de proyección, mientras que el segundo se relaciona con la ejecución final, una vez completada la etapa de construcción, donde se podrá ver que la distribución de las instalaciones ha variado, aunque no de manera significativa.

El concepto principal que gira alrededor de ambos es la convergencia entre la información digital y la construcción real, la visualización realista de los modelos, la comparación de planos (papel y digital), la mejora del trabajo, el etiquetado de los elementos que componen el edificio y, por último, un recorrido virtual.

1.2. Planteamiento del problema

El trabajo se ha realizado empleando el modelo del edificio B0 del Campus Nord de la UPC, lo que nos proporciona una información real y próxima, a partir de la información proporcionada por el arquitecto. Los casos de uso, como se ha dicho previamente, son los siguientes.

A. Caso de uso: Arquitectura y Estructura.

- Fuente: información diseño según formatos de 3ds Max + AutoCAD
- Estudio: Etiquetado de elementos, comparación y visualización de planos y recorrido por la estructura
- Análisis: Unity 3d

B. Caso de uso: Instalaciones.

- Fuente: fichero Revit de modelo con información *as-built*
- Estudio: Visualizar las instalaciones, estudiar el posible solapamiento de trabajos, comparación de planos con el primer modelo y un recorrido virtual.
- Análisis: Unity 3d

1.3. Objetivos

1.3.1. Objetivos generales

El objetivo del trabajo se puede desglosar en los siguientes puntos:

- ✓ Aprendizaje de los principios básicos de la realidad virtual, los conceptos, aplicaciones, dispositivos y herramientas asociadas, como base teórica para el estudio de sus posibilidades
- ✓ Aprendizaje de los principios, procedimientos y herramientas BIM, especialmente en relación con las maneras de trabajar tradicionales

- ✓ Uso de herramientas de realidad virtual y BIM sobre una obra específica, que permita valorar el potencial de estas tecnologías
- ✓ Implementación de casos de uso para la valoración de la realidad virtual en el sector de la construcción, en particular para examinar las ventajas de un recorrido virtual conectado con posibilidades a nivel de las diferentes etapas del proceso constructivo: proyecto, construcción y operación
- ✓ Evaluar desde el punto de vista técnico y económico su posible ejecución en la vida real. Debemos de evaluar hasta qué punto es factible llevar a cabo el proyecto, cuál sería su coste aproximado, cómo se llevaría a cabo y por último y más importante si su uso en realidad traería consigo las mejoras que se buscan.

1.3.2. Objetivos Específicos

De forma específica, este trabajo permitirá examinar aspectos técnicos y tecnológicos tales como:

- ✓ Aprendizaje de los diferentes ficheros de intercambio para los diferentes programas de tratamiento de información digital
- ✓ Comparativa entre las formas de trabajo tradicional y basadas en BIM-RV
- ✓ Anticipar de manera detallada los posibles conflictos que se puedan presentar entre los diferentes proveedores y subcontratistas en toda obra, gracias a la información digital y su visualización común
- ✓ Valorar la dificultad de adopción de la tecnología respecto a los actuales procedimientos de trabajo.

1.4. Metodología

A lo largo de los años el sector de la construcción ha ido experimentando numerosos cambios en cuanto a las formas de trabajo, maneras de ejecutarlo y además con la aparición de nuevas tecnologías nacieron también nuevas herramientas de trabajo.

En la actualidad las constructoras están optando por trabajar con modelos digitalizados ya que estos permiten tener toda la documentación actualizada y ordenada. Con estas filosofías de trabajo nace los entornos BIM, donde el objetivo es reducir los costes y mejorar la eficiencia y eficacia del trabajo.

El foco principal de este proyecto se basa en conectar el modelado BIM con RV partiendo de la premisa de que ambas disciplinas conforman el trabajo de modelado 3D.

Por tanto, el trabajo a desarrollar en este proyecto implica en primer lugar revisar con detalle que es RV explicar cómo podría conectarse con un entorno BIM y finalmente proponer dos casos de uso que sirvieran para conocer el detalle de la tecnología y validar su utilidad en la construcción.

Modelo 1: Caso de uso. Arquitectura + estructura

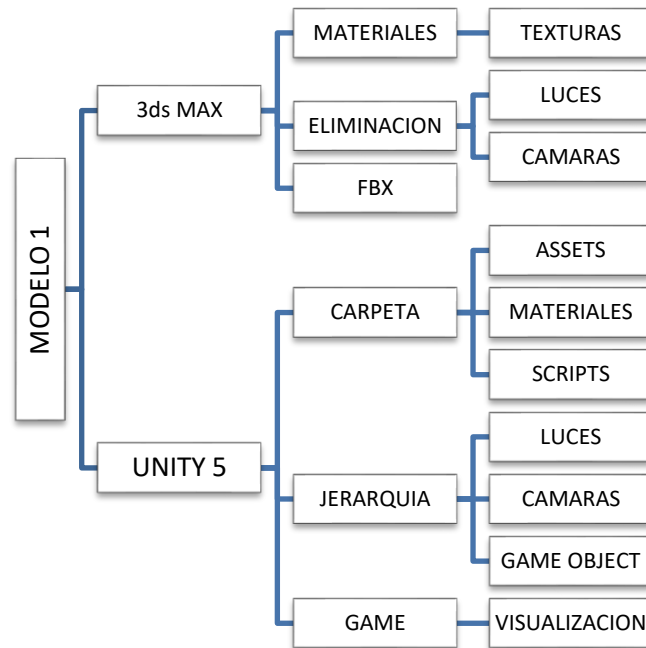


Figura 1. Esquema colaborativo: Caso de uso1 . Arquitectura + Estructura

En este caso de uso vamos a explorar las posibilidades que ofrece el recorrido virtual para la construcción como por ejemplo la revisión de la arquitectura o el etiquetado virtual de espacios.

Modelo 2: Caso de uso. Instalaciones

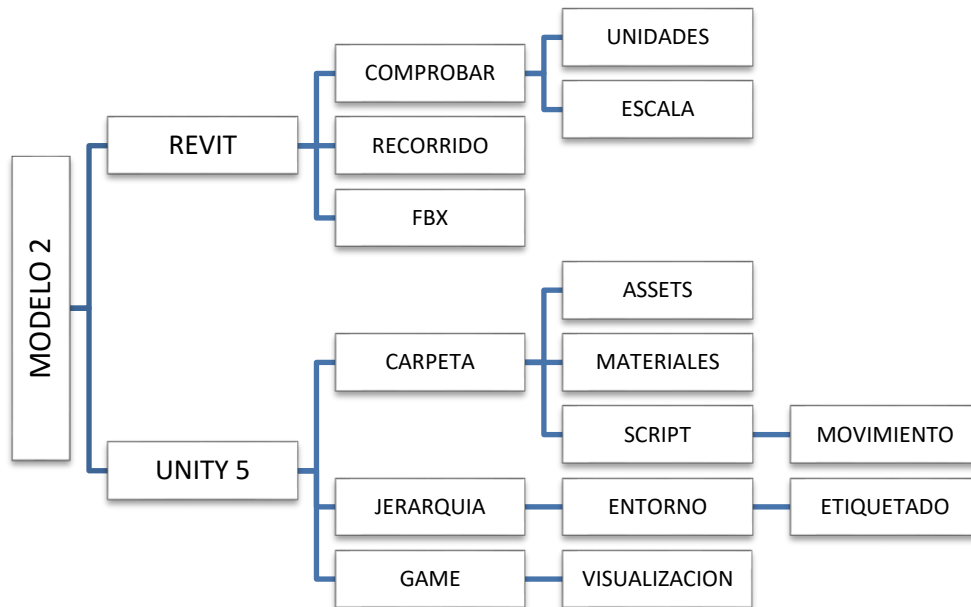


Figura 2. Esquema colaborativo: Caso de uso 2. Instalación

En este segundo caso de estudio nos vamos a centrar en la revisión de las instalaciones como medio para evitar futuros posibles solapamientos en los distintos trabajos que se realizan en obra.

2. Capítulo 2: Estado de la tecnología

A continuación, vamos a mostrar un esquema a modo de repaso de las tecnologías que se van a utilizar a lo largo del desarrollo de este proyecto.

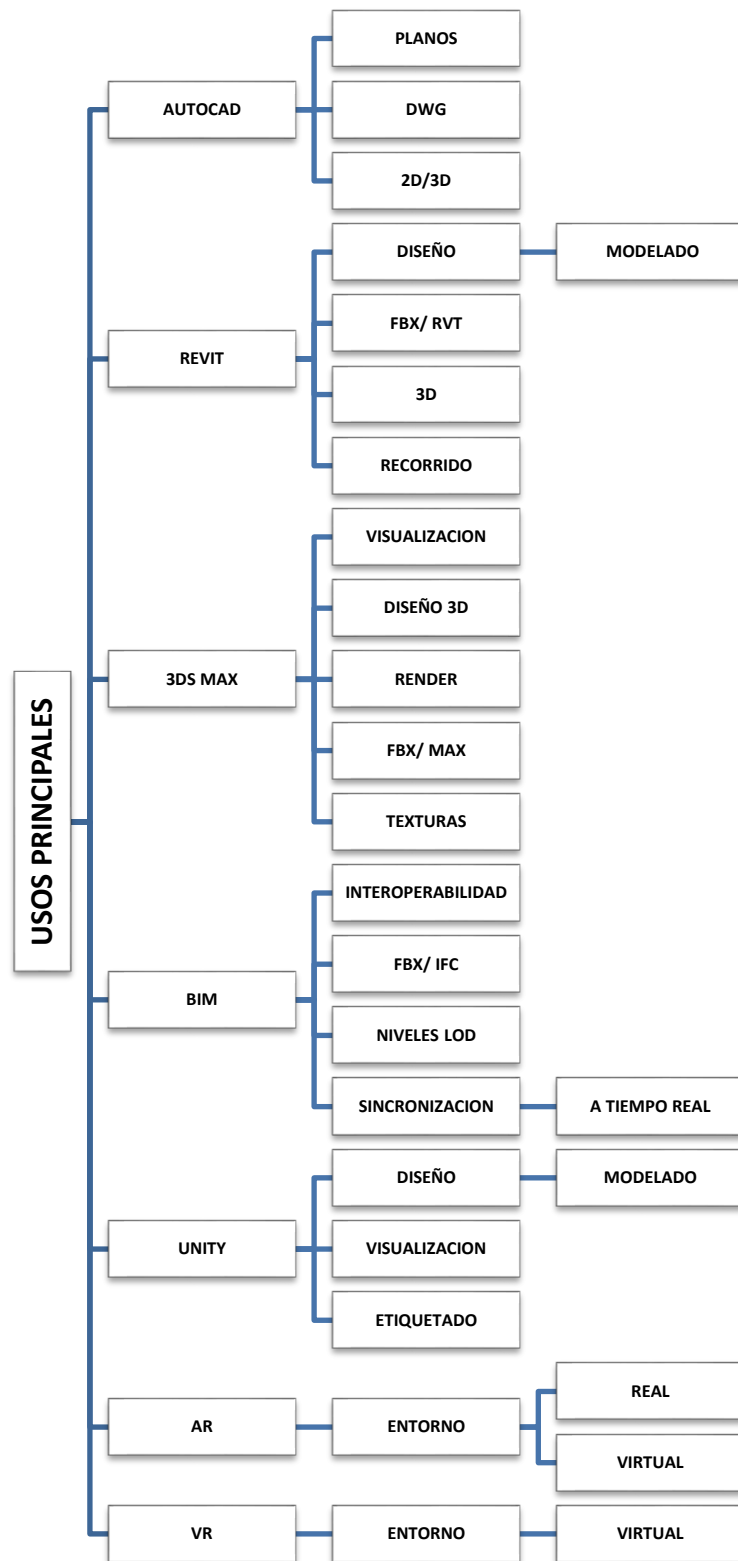


Figura 3. Comparación de los distintos formatos de Trabajo

2.1. BIM

Se define como un modelado de la información para la construcción BIM, viene del acrónimo inglés (Building Information Modeling) nace con el objetivo de digitalizar todos sus elementos a través de una biblioteca donde se guardan todos los elementos inteligentes (son elementos creados de manera exclusiva solo para programas con implementación BIM), es decir, guarda toda la información que sea necesaria para la realización de cualquier obra, como son los muros, pilares, ventanas, puertas, escaleras, etc., ya sea por ejemplo para un edificio de oficinas o un campo de futbol. Convirtiéndose así en un prototipo digital de los elementos físicos del edificio. Esto nos va a poder permitir simular y entender el comportamiento de este en un entorno virtual antes de su construcción real. [1]

Aunque dependiendo del autor o de la página donde se ha realizado la consulta se puede encontrar varios puntos de vista sobre BIM, las características más comunes son: Se define como un software de modelado y visualización en tres dimensiones, lo describen también como un modelo 3D virtual de edificios, otros dicen que BIM es un proceso y, por último, como una colección de datos de un edificio organizados en una base de datos estructural que se puede consultar fácilmente de forma visual o numérica.

Uno de los grandes objetivos que se pretende alcanzar con este tipo de software es disminuir la pérdida de tiempo y los recursos en el diseño y la construcción. Hay que tener en cuenta también que el concepto BIM que se manejaba hace unos años ha ido evolucionado con el tiempo. Actualmente se podría decir que se maneja una estructura en torno a tres pilares que son las personas, los procesos y las tecnologías. Dentro de cada uno de cada uno de ellos, hay unas ideas que son claras e importantes a la vez: la colaboración, el intercambio de información y contrato (LOD) y la digitalización donde se manejan las herramientas en cuanto a la creación de construcciones, análisis y gestión de la información, que maneja el Facility Management y Project Management. [2]

Por otro lado, si miramos la evolución que ha tenido BIM en la Unión Europea, vamos a poder diferenciar dos ideas importantes que son: innovación (Horizon 2020), implantación (Smart Cities) y uso, que van asociado a la quinta dimensión que posee BIM (en la figura 9 hablemos más en detalle de cada una de las dimensiones que posee BIM).

Una de las grandes ventajas que posee es la optimización en el tiempo y los recursos destinados en las fases de diseño y construcción, con una notable mejora en la gestión de explotación de las infraestructuras. La empresa húngara Graphisoft fue la pionera en la aplicación del concepto BIM con el programa ArchiCAD, este software era capaz de crear dibujos en 2D y 3D. Bien si hacemos una comparación con la programación en CAD, se puede ver que estos utilizan sólo geometría en 2D o 3D sin diferenciar los elementos, sin embargo, BIM utiliza bibliotecas, estas son capaces de interpretar la interacción lógica entre los diferentes tipos de objetos y almacena la información referente a estos. Por tanto, este proceso abarca la geometría del edificio, las relaciones espaciales, la información geográfica, así como las cantidades y las propiedades de sus componentes. [3]

Uno de los usos que se puede obtener con BIM es la ilustración de un proceso completo en estructuras de edificación, del mantenimiento e incluso la demolición (ya que ahora se reciclan más materiales). Hoy en día este software es ofrecido por diferentes proveedores tecnológicos como: Tekla, Autodesk, Bentley Systems entre otros [4]

Sin embargo, BIM marca un antes y un después en la nueva era que vivimos, ya que los profesionales de la Arquitectura, Ingeniería y Construcción “AEC” (Architecture, Engineering and Construction), van a poder no sólo ahorrar tiempo al crear y modificar

sus proyectos, sino que también facilitará la interacción al más alto nivel con sus colaboradores, asociados o colegas, al compartir contenidos específicos de cada especialidad en el mismo modelo BIM. [1]

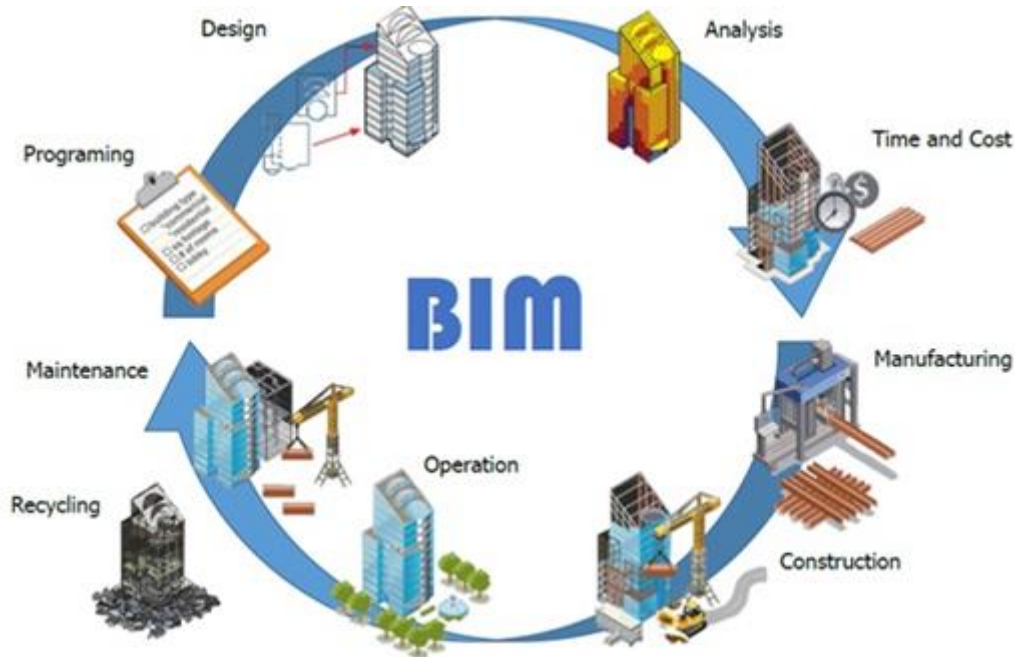


Figura 4. Funcionamiento de un flujo BIM [AMBIM Arquitectura e ingeniería]

2.1.1. Ciclo de vida de un proyecto en BIM

Un proyecto pasa por múltiples fases del ciclo de vida (PLPs⁶). Dentro de cada fase se tendrá en cuenta una sub-fase, las tres fases más importantes son. [5]

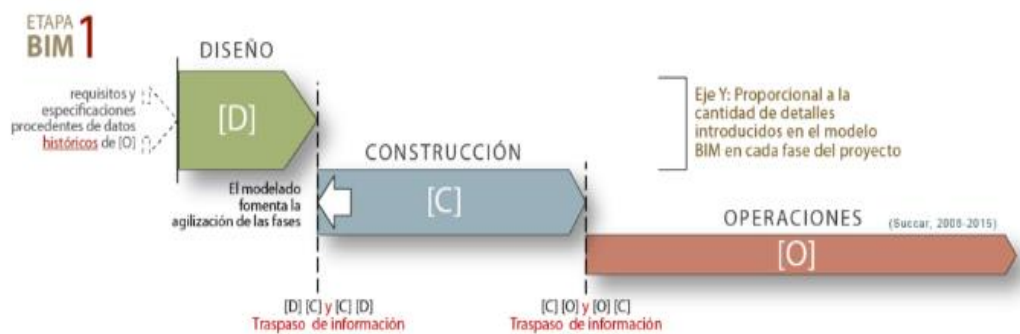


Figura 5. Fases del ciclo de vida del Proyecto en Etapa BIM 1 – proceso lineal

La fase de adaptación al modelo BIM es básica. El 95% del trabajo se realiza con dibujos en 2D donde la poca coordinación incrementa el coste. 25% reelaboración de los trabajos.

[Anxjo 1: Metodología BIM]

⁶ Project Lifecycle Phases

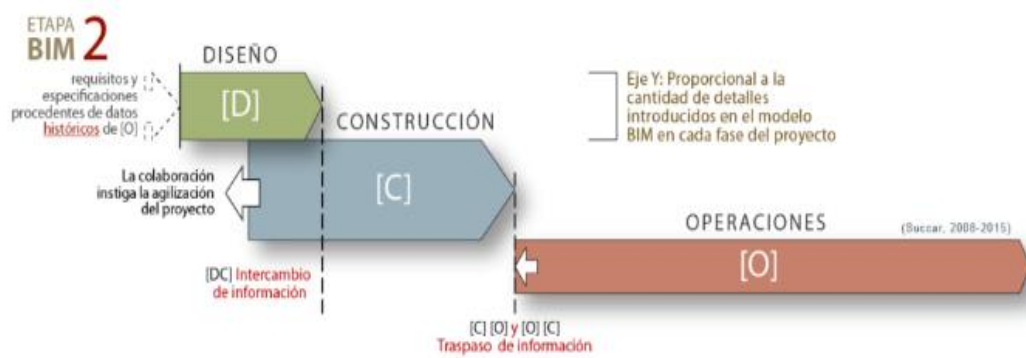


Figura 6. Fases del ciclo de vida del Proyecto en Etapa BIM 2 – modelo lineal

Si llegamos a la segunda fase quiere decir que estamos a medio camino, es decir la coordinación del modelado 2D/3D está basada bajo BS 1992:2007, la posibilidad de error aun es alta donde el 50% del trabajo no termina de ser eficaz



Figura 7. Fases del ciclo de vida del Proyecto en Etapa BIM 3 – modelo lineal

Es la fase más deseable, cualquier empresa que llegue a este punto se encontrará totalmente integrada en la metodología (BIM) ya que sigue la normativa 9SO BIM.

A pesar de categorizar las fases en tres grupos una forma más sencilla de entender y visualizar, es dividiendo en dos bloques:

- Fase Virtual del Edificio Inteligente. Se identifica con la fase de diseño. Es el momento clave para conocer el verdadero alcance que tendrá el edificio y el grado de integración de la tecnología que se aplicará.
- Fase Real del Edificio Inteligente. Se identifica con la fase de ejecución y mantenimiento. Representa una puesta en escena del concepto del edificio. Tras la ejecución y culminación del proyecto, serán los administradores o propietarios quienes se encargarán del mantenimiento

Debe tener en cuenta que la unión fases-BIM a lo largo del proyecto es importante ya que, de otro modo, la probabilidad de equivocación en la fase de ejecución es alta y los costes de bonificación se dispararán. Como consecuencia, se tendría que reorganizar y contratar a un equipo técnico que junte toda la información, estudie el edificio y aporte aquellos datos que nos hubiera aportado un proyecto elaborado desde su planteamiento inicial con BIM. [5]

2.1.2. Ventajas BIM vs método tradicional.

A continuación, vamos a definir las características más relevantes que presenta la metodología BIM con respecto al método tradicional

- Detección previa de los posibles conflictos en las distintas fases del proceso
- Aumenta y optimiza el tiempo
- Estima las repercusiones de cada solución en las distintas áreas, sin improvisar nuevos modelos
- Permite proyectar el calendario del proyecto y realizar un seguimiento durante la construcción, además, facilita el cálculo de la eficiencia energética del edificio y de sus instalaciones.

Una forma más clara de ver es a través del siguiente ejemplo, donde vamos a comparar dos metodologías de trabajo una con respecto al modo tradicional (AutoCAD) y otra adaptado a BIM (Revit).

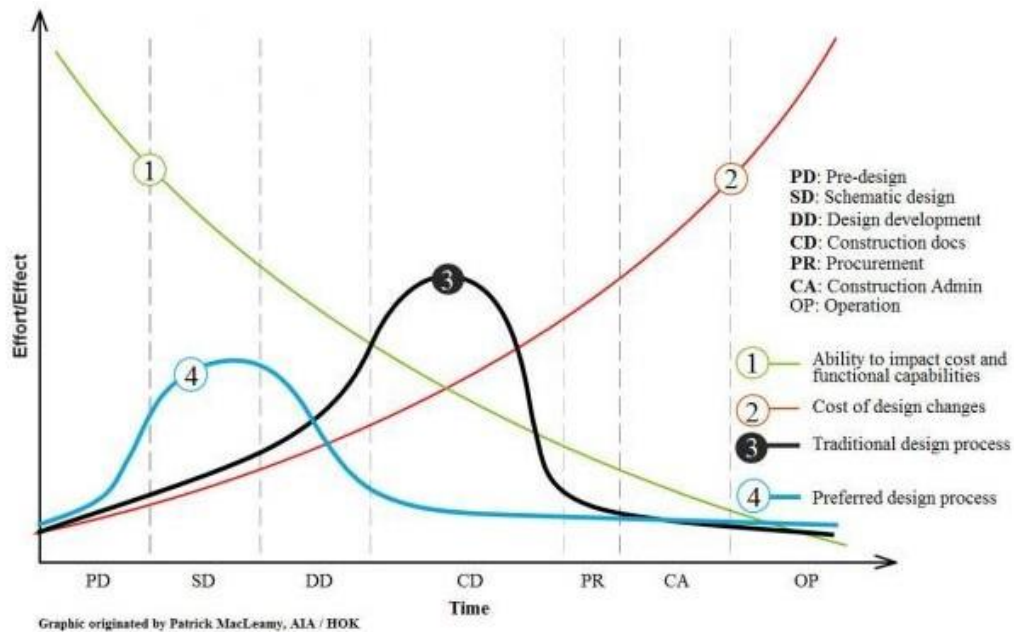


Figura 8. Ventajas en la comparación de trabajo Revit vs AutoCAD [CEO de HOK 2008]

Si analizamos la figura 8 podemos observar que la gran ventaja que posee Revit es la de proporcionar una posibilidad de detección de problemas en el diseño en una etapa temprana dentro del tiempo de desarrollo del proyecto, como consecuencia se tendrá un mayor beneficio en cuanto a productividad y economía. Esto es muy importante ya que a mayor tamaño de proyecto mayor beneficio y productividad.

2.1.3. Clasificación de dimensionamiento

La clasificación de dimensionamiento es la esencia de BIM, ya que marca el alcance de cada dimensión de forma acotada a lo largo del transcurso del proyecto. Otras formas de nombrar dichas dimensiones son como: fases, procesos del proyecto o simplemente como un producto.

La figura 9 muestra tales dimensiones, donde la figura Project Management⁷ va asociada desde la dimensión 3D que corresponde al modelado del proyecto hasta la 6D donde se lleva a cabo el análisis energético del proyecto y por último el 7D donde la figura del Facility Management⁸ es la figura más importante en una sola dimensión. [6]

⁷También es conocido como gestor del proyecto. Es la persona que tiene la responsabilidad total del planeamiento y la ejecución acordados de cualquier proyecto

⁸ Es el profesional que se encarga de diseñar o adaptar los espacios e infraestructuras de una empresa para que los empleados desempeñen sus actividades con la mayor eficacia posible.

Clasificación de dimensionamiento				
3D	4D	5D	6D	7D
<i>Modelado</i>	<i>Calendario</i>	<i>Control de costes</i>	<i>Sostenibilidad</i>	<i>Mantenimiento</i>
Descripción grafica	Simulación de las fases del proyecto	Modelado conceptual en tiempo real	Análisis energético	As Built
Información geométrica	Simulación de instalaciones	Estimación de costes	Variaciones e interacciones de a envolvente	Modelo de operación y mantenimiento
Visualización del proyecto	Diseño de plan de ejecución	Cantidades de materiales	Seguimiento de los elementos sostenibles	Control de logística del Proyecto
Objeto con propiedades		Soluciones de prefabricación	Seguimiento LEED	Ciclo de vida útil

Figura 9. Clasificación del dimensionamiento [BIM Barcelona]

2.1.4. Evolución de cambio de CAD a BIM

BIM y CAD son dos formas de interpretar diferentes maneras de trabajo en cuanto al diseño arquitectónico y la documentación. Si hablamos de CAD el diseño sigue un modelo paramétrico que imita el trabajo que se solía hacer con papel y lápiz. Se crean de manera independiente, donde cualquier cambio que se realice en el diseño se deberá revisar en todo el proyecto y modificar manualmente cada cambio. En cambio, BIM no construye dibujos en 2D, imita el proceso real de construcción a través de la biblioteca donde se guarda todos los materiales, tales como muros, ventanas, forjados, cubiertas, etc. Esto permite a arquitectos diseñar edificios de la misma forma en que son construidos ya que como todos los datos están guardados en un modelo central (se trata de un trabajo en red donde toda la información se comparte), los cambios en el diseño son automáticos consiguiendo así el incremento de la productividad y mejorando la coordinación. [7]

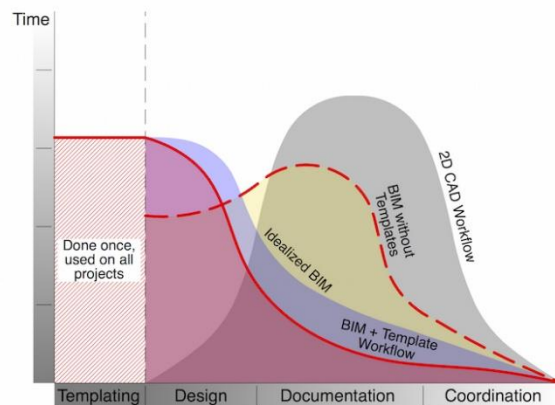


Figura 10. Optimización de los recursos BIM vs CAD [Arquirapados]

La figura 10 nos muestra varios flujos de trabajo, donde una de las diferencias se puede apreciar en la primera fase (Templating), donde el tiempo de configuración se ve notablemente reducido ya que los beneficios están en curso, eliminándose drásticamente el tiempo que tarda en iniciar el proyecto, donde la documentación y coordinación con caso automáticos (50-80%)

2.1.5. Niveles de desarrollo según el marco europeo

A pesar de que el trabajo desarrollado con BIM tiene importantes ventajas, a la hora de definir el coste del modelo nos encontramos ante un problema ya que valorar lo que aporta cada parte al proyecto es complicado. Atendiendo a este problema se crea lo que en un principio se denominó nivel de detalle LOD (Level Of Detail). *Se corresponde a la evolución lineal de cantidad y riqueza de información de un proceso constructivo; siempre aumenta con el tiempo y se refiere al modelo de proyecto, los costes/presupuestos y la planificación temporal*, pero conforme fue evolucionando el modelo la AIA (American Institute of Architects) decidió que sería una buena oportunidad el también valorar la calidad del modelo, por tanto, aquí nace el concepto que actualmente se conoce como nivel de desarrollo LOD (Level Of Development). *Se define como el nivel de desarrollo o madurez de información que posee un elemento del modelo, es decir, es la parte de un componente, sistema constructivo o montaje del edificio*. Hay que destacar que en ningún caso se refiere al total del proyecto y tampoco tiene vinculación alguna con la fase de construcción o plan de ejecución BIM (BEP⁹). Todos los niveles están desarrollados o categorizados por: [2] [5] [8]

Nivel de desarrollo (LOD)				
LOD 100	LOD 200	LOD 300	LOD 400	LOD 500
<i>Anteproyecto</i>	<i>Proyecto básico</i>	<i>Proyecto previo a la ejecución</i>	<i>Proyecto de ejecución</i>	<i>Planos As - built</i>
Es un diseño conceptual, el modelo aportará una visión general, básicamente aportará el volumen, la orientación y área	Aportará una visión general con información de magnitudes aproximadas, tamaño, forma, localización y orientación. El uso que se da es simplemente aumentar la capacidad de análisis	El elemento de modelo se representará gráficamente en el modelo como un sistema específico, objeto o ensamblaje en términos de cantidad, tamaño, forma, ubicación y orientación. La información no gráfica también se puede incluir en el modelo del objeto	Contiene el detalle necesario para la fabricación o construcción. Donde el nivel de decisiones es exacto.	Este último nivel de desarrollo representa el proyecto, ya que se ha construido, con las condiciones conforme a obra. El modelo es adecuado para el mantenimiento y funcionamientos de la instalación.

Figura 11. Niveles de Level Of Developed en España [ARUP España] [3]

⁹ BIM Execution Plan

2.1.6. Formato universal IFC

Industry Foundation Classes (IFC), es un formato de archivo nacido con el propósito de convertirse en un formato estándar, que facilite la interoperabilidad entre softwares. Permite compartir información y evitar la pérdida de datos al pasar de un programa a otro. El formato de los datos es de especificación abierta

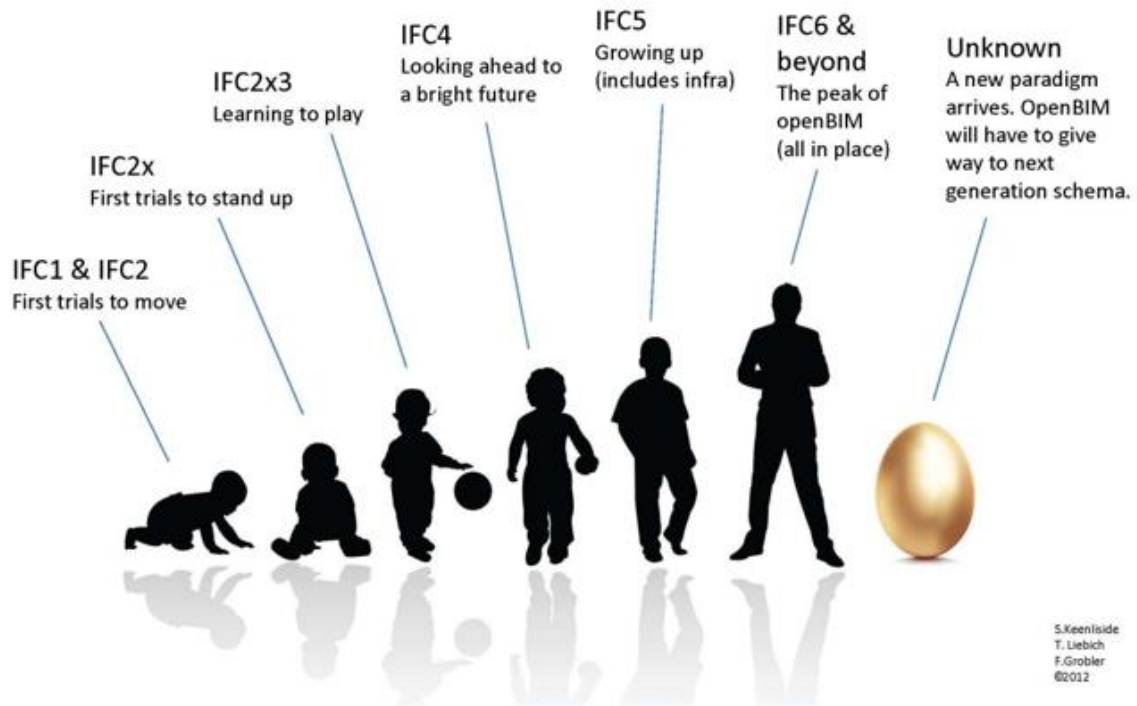


Figura 12. Desarrollo del nivel de madurez de un IFC [Keenlside, Liebich&Grobber 2012]

En la figura 12 se puede ver las distintas etapas por las que ha ido pasando a lo largo del diagrama de madurez, es decir nos muestra su evolución y crecimiento hasta llegar a la madurez, momento en el cual es reemplazado o incluido en el siguiente método. Hay que destacar que IFC todavía tiene mucho que ofrecer en futuras iteraciones [1]

2.2. Revit

Revit fue creado de forma exclusiva para trabajos en modelado BIM (Building Information Modeling) y está desarrollado por Autodesk. Este software permite al usuario diseñar con elementos de modelación y dibujo paramétrico. Se trata de dibujo asistido por ordenador que permite un diseño basado en objetos inteligentes y en tres dimensiones una de las características principales que posee Revit y que le distingue de otros programas es que un cambio en algún lugar de cualquiera de sus componentes significa un cambio en todos los lugares, instantáneamente, sin la intervención del usuario para que este cambie de manera manual todas las vistas. Un modelo BIM debe contener el ciclo de vida completo de la construcción, desde el concepto hasta la edificación. Para hacer esto posible se apoya en un software de diseño que integra todas las tareas a realizar. [9]

Las funciones de este software cubren las necesidades de modelado tanto de la arquitectura, ingeniería y la construcción. Algunas de las más importantes son: [10]

1. **Componentes de diseño y construcción:** El programa cuenta con herramientas para diseñar el edificio desde su conceptualización hasta la planimetría de la construcción. Esto abarca detalles en muros, pisos, cielos y cubiertas, incluyendo los muros cortina. Además, permite realizar un estudio volumétrico mediante masas, calcular áreas por pisos y experimentar con texturas, materiales y colores, entre otras aplicaciones.
2. **Sombras Vectoriales:** Al realizar cualquier cambio en la estructura, orientación y otros detalles que modifiquen la disposición de elementos frente a la luz, las sombras se ajustan inmediatamente, permitiendo visualizar el efecto de los cambios en la iluminación.
3. **Perspectivas seccionales:** Permite analizar todos los ángulos del edificio desde distintas perspectivas y en distintas secciones, incluyendo vistas con líneas ocultas, sombras y siluetas.
4. **Modelo de proyecto integrado:** Posee un conjunto de herramientas para coordinar las distintas áreas del proyecto, sus documentos e información relacionada. Produce referencias automáticas de dibujo, estima costos, permite modificar la geometría solo al ingresar números, coordina las versiones para que todos los datos, gráficos, detalles y dibujos estén actualizados en todas partes, entre varias otras funciones orientadas a optimizar los tiempos y mejorar la calidad de las entregas.
5. **Modelado de terreno y exteriores:** Permite diseñar el edificio tomando en cuenta el contexto exterior, entregando diseños de pisos y patrones. También ofrece una biblioteca con vegetación y otros elementos, como la maquinaria de construcción, para planificar los procesos de construcción de forma adecuada.
6. **Ambiente de trabajo multidisciplinario:** Los distintos equipos pueden trabajar de forma simultánea en un edificio y el programa coordinará todos los cambios ingresados.
7. **Presentación y visualización:** Cuenta con renderización integrada que incluye puertas, ventanas y tragaluces en sus cálculos para simular las condiciones de luz natural, entre varias otras funciones. También realiza análisis de área para producir esquemas, tiene un pantone integrado y permite exportar a PDF para imprimir o enviar vía email.

2.3. Unity 5

Unity 5 [11] [12] es un motor y marco 2D/3D que brinda un sistema para diseñar escenas de juegos o de aplicaciones. no solo se puede ver en el sector de los videojuegos, sino también en simuladores de aprendizaje, aplicaciones para socorristas entre otras. Unity 5 permite interactuar con ellas no solo a través de códigos, sino también mediante componentes visuales, y exportarlos a cualquiera de las principales plataformas móviles además admite todas las aplicaciones 3D principales y muchos formatos de audio, incluso es compatible por ejemplo con el formato de Photoshop. Unity 5 permite importar y crear activos, escribir código para interactuar con objetos, crear animaciones.

La parte más eficaz de Unity 5 es la Asset Store, allí se puede encontrar todo para sus necesidades de componentes para juegos, como material gráfico, modelos 3D, archivos de animación para los modelos 3D, efectos de audio y pistas completas, complementos, incluidos algunos como el kit de herramientas multiplataforma, que puede ayudar con la compatibilidad para varias plataformas, sistemas de scripting

visual, sombreadores avanzados, texturas, efectos de partículas, etc. La interfaz de Unity permite ejecutar scripts en su totalidad, lo que permite integrar complementos de terceros directamente en la GUI de Unity. [13] [14]

Unity no es un sistema en el que pueda diseñar activos 2D y modelos 3D (excepto para terrenos), como puede ser Autodesk Maya, 3ds Max o Blender. [13]

2.4. Realidad virtual

La realidad virtual RV (Virtual Reality VR) se puede definir de una manera o de otra, todo depende del autor donde se consulta, a pesar de todo, todas ellas guardan una línea de semejanza.

- Según Manetta C. y R. Blade (1995) *la realidad virtual es un sistema computacional usado para crear un mundo artificial en el cual el usuario tiene la impresión de estar y la habilidad de navegar y manipular objetos en él.*
- Según Hodder y Stoughton, *la realidad virtual permite explorar un mundo generado por ordenadores a través de nuestra presencia en él.*
- Según Aukstakalnis (1992) *la realidad virtual es un camino que tienen los humanos para visualizar, manipular e interactuar con ordenadores una información extremadamente compleja.* [15]

Basándonos en estas tres ideas se podría decir que la Realidad Virtual se define como un sistema tecnológico que pretende simular las percepciones sensoriales de forma que el usuario las tome como reales. Para ello se define lo virtual como algo que percibimos pero que no se corresponde con la realidad en ese espacio-tiempo (espejismo, película o sueño). Si se quiere que el usuario perciba algo virtual como real se necesitara una interfaz que lo simule en tiempo real y le permita interactuar con él a través de los múltiples canales sensoriales (visión, audición, tacto, olor y gusto). Sin embargo, la mayoría de los sistemas actuales se centran en únicamente 2 sentidos (vista y oído), debido a la dificultades y costes de simular los otros sentidos. [16]

Esta realidad además puede ir acompañado por otros dispositivos, como guantes o trajes especiales, que permiten una mayor interacción con el entorno, así como la percepción de diferentes estímulos que intensifican la sensación de realidad. La aplicación de la realidad virtual se centra inicialmente en el terreno del entretenimiento y de los videojuegos, aunque se ha ido extendiendo a otros muchos campos, como la medicina, la arqueología, la creación artística, el entrenamiento militar, las simulaciones de vuelo o, en mi caso de estudio para el sector de la construcción, por ejemplo. [17].

El objetivo principal no es otro que crear, almacenar y simular un mundo alternativo, modelar objetos en él, definir relaciones entre ellos y la forma en la que interactúan, para que el usuario pueda más tarde percibirlo.

1. Las tres “I” de la realidad virtual.

- **Inmersión:** El usuario pierde contacto con la realidad al percibir únicamente los estímulos del mundo virtual
- **Interacción:** El usuario interactúa con el mundo virtual a través de dispositivos de entrada, de forma que modifica cosas en él y recibe la respuesta a través de sus sentidos (tiempo virtual = tiempo real).
- **Imaginación:** A través del mundo virtual podemos concebir y percibir realidades que no existen, de manera parecida a como hacemos con la creación artística. [15]

2. Tipos de visualización

- **Inmersiva:** Se basa en la simulación de un ambiente tridimensional el cual el usuario percibe a través de estímulos sensoriales (cascos de realidad virtual o gafas), hasta el punto de desaparecer el mundo real.
- **No inmersiva:** Opta por la visualización de los elementos virtuales por una pantalla, dando opción de interaccionar con otras personas a través de Internet. Normalmente la tecnología de VR no inmersiva es más barata y más familiar para el usuario. Este tipo de VR es muy común en videojuegos en la actualidad ya que no requiere ningún hardware especial [15] [16]

3. Relación real/ irreal:

- La realidad virtual es compartida por otras personas, que interaccionan sin estar en el mismo espacio-tiempo.
- Tiene una estrecha relación con el mundo físico debido a su interrelación e influencia mutua.
- Está influenciada por la creación artística, ya que lo que percibimos a través de los dispositivos debe ser creado por una o varias personas. [15]

4. Elementos de un sistema de realidad virtual.

- **Entrada de datos:** Realiza un seguimiento del usuario y su interacción con el entorno virtual (guantes de datos, ratones 3D o trackers).
- **Salida de datos:** Realimenta los dispositivos sensoriales del usuario: sonido, vídeo (HMD), tacto (guantes de datos).
- **Motor de realidad:** Máquina que alberga el software donde se creará el mundo virtual. Necesita un hardware acorde con la calidad de VR que se requiera. Normalmente se trata de estaciones gráficas de gran desempeño.
- **Software RV:** Lenguajes, librerías y sistemas autorizados que se usan para implementar interfaces completas para diferentes mundos virtuales.
- **Base de datos del mundo:** Contiene los objetos del mundo virtual y sus propiedades.
- Además, el motor de Realidad Virtual puede estar basada en un modelo cliente-servidor de la siguiente forma:

Servidor:

- ✓ Sincronización de las imágenes.
- ✓ Control de colisión de las imágenes.
- ✓ Comunicación con los clientes, que visualizarán el mundo virtual.

Cliente:

- ✓ Contiene el mundo virtual.
- ✓ Se comunican con el servidor.

Cada cliente podrá tener perspectivas distintas del mundo, mientras que el servidor enviará órdenes de navegación para que los clientes actúen en consecuencia. [15]

5. Interacción

- **Humano – maquina:** Únicamente puede interaccionar una persona por mundo virtual. Ejemplo de este tipo de VR serían los videojuegos no multijugador.
- **Humanos – maquina:** Es posible que más de una persona compartan el mismo mundo virtual e interacción al mismo tiempo con el mismo y/o entre ellos. [16]

6. Clasificación de los mundos virtuales

- **Mundo muerto:** No hay interacción entre el sistema y el usuario ni movimiento de los objetos. El usuario percibe un mundo virtual a través de los sentidos, pero no puede actuar sobre él, sólo explorarlo. Por ejemplo, las animaciones tradicionales de imágenes pre-calculadas.
- **Mundo real:** Los objetos poseen los atributos correspondientes a su equivalencia real, de manera que podemos interactuar con ellos tal y como hacemos en el mundo real. Si miramos un reloj, éste marca la hora, y si arrastramos un lápiz por el papel, escribimos. Por ejemplo, un simulador de conducción dentro de una ciudad.
- **Mundo fantástico:** En él podemos realizar tareas irreales, como volar o atravesar paredes, abriéndonos más posibilidades de exploración del mundo. Por ejemplo, como los mundos mágicos o los superhéroes. [15]

2.5. Realidad aumentada

Se puede definir como un incremento de la información que el ser humano puede obtener al interactuar con el mundo físico y digital en tiempo real y en el mismo espacio. De esta manera, se puede obtener mayor información de los elementos que conforman nuestro entorno, es decir, la realidad aumentada busca ampliar nuestra percepción mediante imágenes, vídeos o información digital con ayuda de dispositivos móviles, como Smartphone, Tablet, ordenadores o las Google Glass, convirtiendo así nuestro mundo físico en interactivo y digital. En la actualidad la realidad aumentada tiene muchas utilidades tanto en el sector de la medicina como inmobiliario o en el ámbito de la construcción. Para acceder a esta información solo se debe de tener algún dispositivo de los nombrados previamente. [18]

1. Características:

- Capacidad de integrar información virtual dentro de una escena real de un modo realista e intuitivo, y en tiempo real. La aplicación práctica más extendida de la realidad aumentada consiste en la superposición de texto e imágenes sintéticas sobre la imagen real capturada por una cámara
- El máximo provecho de la realidad aumentada se obtiene cuando se pueden asociar de forma muy precisa elementos reales con elementos virtuales, para lo cual es necesario utilizar técnicas como la visión por computador, que permiten realizar seguimiento de objetos en tiempo real
- Las técnicas utilizadas se basan originalmente en la detección de marcas artificiales prediseñadas colocadas de ante mano sobre los objetos de la escena que queremos identificar
- La realidad aumentada combina información real y virtual.
- La realidad aumentada es tiempo real
- Es interactiva en tiempo real. [19]

2. Componentes

- Monitor: Instrumento donde se verá reflejado la suma de lo real y lo virtual que conforma la realidad aumentada
- Cámara web: Dispositivo que toma la información del mundo real y lo transmite al software de realidad aumentada
- Software: Programa que toma los datos reales y los transforma en realidad aumentada

3. **Arquitectura:** Será necesario un dispositivo que capture toda la información sobre la realidad real, de forma que pueda procesarse la información que ésta contiene, se usarán cámaras de vídeos en algunas arquitecturas y un sistema para generar las imágenes virtuales que se quieren implementar con el fin de aumentar la realidad
 - Lentes reflectantes (sistema óptico)
 - Cascos con monitores (sistema de vídeo)
 - Monitores (sistema de vídeo)
4. **Funcionalidad:** En la actualidad, los servicios de realidad aumentada se configuran generalmente incrustando pequeños fragmentos de información digital en la realidad que observan los usuarios. Esta información suele consistir en etiquetas de texto describiendo un determinado objeto o imágenes. Esto debe considerarse como el principio de una tendencia de mezclar información real y virtual que todavía tiene un largo recorrido por delante. El avance de las tecnologías en las que se apoya la realidad virtual marcará el ritmo, tanto para el enriquecimiento de la información digital como de la mezcla de forma transparente con la información que los sentidos nos muestran, creando experiencias de usuario diferentes a las actuales. La realidad aumentada se trata de una aproximación desde la realidad, lo que viene a suponer que la información real predomina y la información virtual. [19]

2.6. 3ds Studio Max

Es un programa de creación de gráficos y animación 3D desarrollado por AutoDesk. Su arquitectura se encuentra basada en plugins, en la actualidad es uno de los programas de animación 3D más utilizados, especialmente para la creación de videojuegos, anuncios de televisión, en arquitectura o en películas, además dispone de una sólida capacidad de edición [20] [21]

1. **Definición de 3D:** En términos computacionales las tres dimensiones son el largo, el ancho y la profundidad de una imagen. Técnicamente hablando el único mundo en 3D es el real, ya que el ordenador sólo simula gráficos en 3D, para crear animaciones, gráficos, películas, juegos, realidad virtual, diseño, etc.
2. **Definición de open GL:** Se definen como un conjunto de especificaciones estándar que definen una API multilenguaje y multiplataforma para escribir aplicaciones o juegos que producen gráficos en 3D.
3. **Definición de DIRECT 3D:** Un API, parte de Direct X, para la programación de gráficos 3D, facilita el manejo y dibujo de elementos en tres dimensiones: líneas, polígonos, texturas, transparencias, etc.
4. **Imagen 3D:** Es el resultado final del proceso de renderizado de un modelo en 3D. Una imagen en 3D, en definitiva, es una imagen en dos dimensiones que simula las tres dimensiones, pero proviene de un mundo conceptual en 3D. Ese mundo en 3D permite que puedan generarse múltiples imágenes en 3D desde diferentes perspectivas. Una imagen en 3D puede almacenarse en cualquier formato gráfico raster. Múltiples imágenes en 3D constituyen una animación 3D.
5. **Animación en 3D:** Una animación 3D hace referencia a un tipo de animación que simula las tres dimensiones, es decir, se trata de la descripción de los objetos de un modelo 3D a lo largo del tiempo. Para que exista animación, esa descripción debe variar en algo con respecto al tiempo. Puede tratarse de una animación que se renderiza en tiempo real cuando se está ejecutando o una animación que utiliza 3D, pero ya ha sido renderizada previamente, por lo tanto, sólo se trata de un video. La principal diferencia entre ambas radica en el momento de renderizado

de la animación, es decir, el proceso de convertir las fórmulas matemáticas en imágenes digitales.

6. **Modelado en 3D:** Desde un punto de vista técnico, se define como un grupo de fórmulas matemáticas que describen un "mundo" en tres dimensiones. Pero desde un punto de vista visual, se define como una representación esquemática visible a través de un conjunto de objetos, elementos y propiedades que, una vez procesados (renderización), se convertirán en una imagen en 3D o una animación 3D. Además, suelen contar con herramientas para la generación de efectos de iluminación, texturizado, animación, transparencias entre otros. El modelo en 3D describe un conjunto de características que, en conjunto, resultarán en una imagen en 3D.
7. **Renderizado:** Es el proceso de generar una imagen (imagen o una animación en 3D) a partir de un modelo, usando una aplicación. Donde el modelo es una descripción en tres dimensiones de objetos en un lenguaje o estructura de datos estrictamente definidos. El modelo debería contener geometría, punto de vista, textura e información de iluminación. La imagen resultante del renderizado es una imagen digital. La renderización se utiliza en la producción de imágenes en 3D. Hay que diferenciar entre pre-renderizado que es un proceso computacional intensivo que es utilizado generalmente para la creación de películas con un resultado de alta calidad y el renderizado que va en tiempo real es más usado en los juegos en 3D. otra definición alternativa es que también puede ser usado para describir el proceso del cálculo de los efectos en la edición de archivos de videos para producir una salida final de video.
8. **Motor de renderizado:** Es el que se encarga de interpretar y representar contenidos. Puede ser un Motor de renderizado web o un motor de renderizado 3D, etc. El motor de renderizado web (rendering Engine), es un software que permite mostrar en pantalla contenido con estilo, estructura y forma de acuerdo a contenido marcado e información de formateo. [22]

Para terminar a modo de resumen hay que decir que un proceso de transformación de un modelo en 3D hacia una imagen 3D es llamado renderización (rendering). El resultado de una renderización puede ser una imagen 3D estática o una animación 3D. [22].

2.7. Realidad aumentada – BIM

Como se ha visto en el apartado 2.4 y 2.5 la realidad virtual y aumentada son dos términos distintos, donde la realidad aumentada, nos proporciona información a tiempo real de actividades reales que se están llevando en cada momento. Si esta herramienta la importamos con un software de modelización, con el ecosistema de modelado BIM, siendo generales, podemos anticipar a cuando se tenga el modelado del edificio y colocado todos sus componentes inteligentes cuando, la persona física realice una visita a través de su dispositivo móvil o a través de unas gafas, va a poder interactuar y ver entre otras cosas el avance de la obra, la maquinaria utilizada, los trabajos que se están realizando en cada momento o, por último, si hay alguna situación de riesgo.

Otro claro ejemplo de la combinación de BIM - AR se puede verse en la figura del Facility Management ya que este podría introducir la vida útil de las lámparas o visualizar las instalaciones. Además, se puede visualizar todos los elementos constructivos que forman parte de la obra, como su situación planos de montaje, posibles incidencias y como subsanarlas en tiempo real o facilitar la ubicación de recursos y planificación de obra y control de costes. [5]

3. Capítulo 3: Caso de estudio

Una vez afianzados todos los conocimientos teóricos para entender o comprender el funcionamiento de las distintas herramientas y metodologías de trabajo que se van a emplear en este proyecto, voy a pasar a explicar la parte práctica, donde se va a explorar, estudiar y analizar, hasta qué punto se puede llegar a conseguir lo propuesto en el capítulo 1 apartado 1.2. Para ello se han creado los casos de uso que se definirán en los apartados 3.1. y 3.2.

Previa a la explicación, he creído relevante crear 5 fichas técnicas con las características más importantes de cada uno de los programas y de los dos casos de uso.

A. Ficha técnica. Caso de uso 1: Arquitectura + Estructura


Ficha técnica: Caso de uso 1		
Ambiente	Nocturno	
Edificio	Campus Nord UPC	
Lenguaje de programación	C Sharp	
Software de cálculo y diseño	AutoCAD	
	3ds Max	
	Unity 5	
Elementos de estudio	Visualización de la arquitectura	
	Evaluación de la estructura	
	Comprobación de planos	
	Etiquetado de elementos	
	Tipo de iluminación	
	Recorrido virtual por sus instalaciones	

Figura 13. Ficha técnica. Caso 1

B. Ficha técnica. Caso de uso 2: Instalaciones


Ficha técnica: Caso de uso 2		
Ambiente	Diurno	
Edificio	Ed. B0. Campus Nord. CIMNE	
Lenguaje de programación	C Sharp	
Software de cálculo y diseño	AutoCAD	
	Revit	
	Unity 5	
Elementos de estudio	Visualización de la arquitectura y estructura	
	Evaluación de las instalaciones	
	Evaluación de la posición solar a lo largo del día	
	Inspección visual de la estructura vista (tuberías, conductos... etc.)	
	Comparación de modelos, con respecto al caso 1	
	Recorrido virtual por sus instalaciones	

Figura 14. Ficha técnica. Caso 2

C. Ficha técnica: Características específicas de 3ds Max

Características específicas de 3ds Max					
Características	Tipo	Versión del modelo	Versión de mi ordenador	Problemas	Efectos
Programa	3ds Max	Creado en 2015	Abierto en 2017	Incompatibilidad	Perdida de luces
					Perdida de cámaras
Motor de render	V-ray		Abierto en 2017	Incompatibilidad	Perdida de materiales
	Mental Ray	Creado en 2015			
Escala		1:100	1:100		
Carpeta de materiales	De 3ds Max				Compatibilidad en Unity 5
Formato de guardado	3ds Max				. MAX
Formato deseado	3ds Max – Unity 5				. FBX
Propósito	Poder trabajar con ambos modelos para contrastar la información de la ficha técnica: Caso de uso 1				
Planteamiento	Preparar el modelo de 3ds Max para abrir en Unity 5				
Objetivo	Exportar el modelo para trabajarlo en Unity 5				
Resultado	Conseguido				

Figura 15. Ficha técnica: Características específicas de 3ds Max

Comprobaciones previas

Versión del proyecto: Como se puede ver en la figura 15, existe una incompatibilidad de versiones, esto nos supondrá un inconveniente ya que, por ejemplo, si un archivo está diseñado en la versión 2017 y se intenta abrir en una versión 2015 no será posible debido a los cambios paramétrico propios del programa.

Pero si sucede lo contrario, es decir abrir un archivo 2015 en una versión 2017 pueden suceder dos cosas:

1. No se podrá abrir por incompatibilidad
2. Es posible abrir, pero se han perdido propiedades del proyecto (Mi caso)

La pérdida de propiedades a su vez puede ser debido a por dos factores, que se encuentran vinculados a los motores de render que posee el propio programa.

- A. *V-ray*: Tiene una gran potencia de acabado en interiores sobre todo cuando parte de la escena está en primer plano, y se aprecian con detalle las texturas e iluminación de los materiales.
- B. *Mental ray*: Trabaja bien en acabados exteriores, parte de una gran ventaja que es su precio y de que no dispone de tantos parámetros como el anterior.

Características

1. El proyecto fue diseñado en la versión 2015 y nosotros realizamos el trabajo en la versión 2017
2. El proyecto fue creado con Metal ray y en mi caso trabajo con V-ray

Escala del modelo: La escala en la que fue creado es la 1:100

Dificultades: Incompatibilidad de formatos, debido a la versión del programa

Evolución del proceso: Aunque las pérdidas que se ha producido es importante, para nuestro caso no será tan relevante ya que partimos con una ventaja y es que tenemos la carpeta de materiales que se creó junto al proyecto, por tanto, llegado el momento lo único que haremos será guardar esta carpeta en una carpeta específica de Unity 5 que se definirá en la figura 16. Lo más importante para nosotros será por tanto valorar que elementos de diseño son realmente importantes y cuáles no, ya que a la hora de vincular el modelo con Unity 5 tenemos que ser capaces de lograr la máxima optimización del modelo para tener una fluidez en el recorrido virtual y además para una futura hipotética vinculación a un dispositivo móvil o gafas.

Por tanto, vamos a eliminar todos aquellos elementos no importantes como:

- ✓ Cámaras
- ✓ Luces

Hay que decir que no se elimina porque sí, todo tiene una lógica y en este caso es debido a que Unity 5 por defecto posee una cámara que va asociada al recorrido con una calidad gráfica superior a la que posee 3ds Max y con una amplia creación de luces que proporcionan un mayor efecto de realismo a la escena.

Resultados: Una vez eliminado todos los elementos secundarios, la siguiente pregunta que nos hacemos es en que tipo de formato debo guardar el archivo para que Unity 5 sea capaz de abrir. El formato de 3ds Max por defecto es MAX, pero este no es válido, una alternativa es guardarlo como FBX (esto nos permitirá mantener la fidelidad total y la funcionalidad del archivo original al vincularla con cualquier otro programa, en este caso Unity 5).

D. Ficha técnica: Características específicas de Unity 5

Características específicas de Unity 5						
Características	Tipo	Form ato	Carpetas	Ruta	Creación de elementos	Objetivo
Vinculación previa	3ds Max	. FBX	Creamos un nuevo proyecto			
Exportar			Texturas	Copiar en Asset		
Escala		1:100				
Crear			Escena			Mantener un orden en la escena de trabajo
			Materiale s			
			Script			
			Estándar Asset		FPS	
Características propias del proyecto					Asignación de materiales a los objetos	
					Ambiente	
					Iluminación	
					Etiquetado	
Lenguaje de programación	C Sharp		Script		Apertura de puertas	Aportar sensación de realismo a la escena
				Nombre de objetos		
				Colisión		
				Recorrido		
Propósito	Poder trabajar con ambos modelos para contrastar la información de la ficha técnica: Caso de uso 1 y Caso 2					
Planteamiento	Preparar el modelo para su visualización, comprobación y recorrido					
Objetivo	Estudiar los elementos de estudio propuestos en la ficha: Caso de uso 1 y caso de uso 2					
Resultado	Conseguido					

Figura 16. Ficha técnica: Características específicas de Unity 5

Características previas:

1. Creado el archivo en el formato FBX, le guardo por ejemplo en el escritorio.
2. Ahora vamos a abrir Unity 5 y creamos un proyecto nuevo, en mi caso lo he llamado Arquitectura
3. Finalizada la creación observamos que:
 - A. Unity 5 genera por defecto una copia de seguridad y una carpeta llamada Asset (esta se encarga de guardar o copiar archivos que se quiera utilizar en el proyecto)
 - B. La estructura organizativa se divide en 5 sub-ventanas

Estructura organizativa en unity 5	
Escena	Sirve para seleccionar y posicionar entornos, como, por ejemplo, el entorno, la cámara o los Game Object (Son los encargados de crear los efectos a través de la definición de unas propiedades características) Es una de las funciones más importantes del programa.
Jerarquía	Contiene cada Game Object de la escena actual. Algunos tienen una conexión directa con los archivos de Asset, como modelos 3D o los Prefabs (objetos personalizados). Es posible reordenar los Game Object arrastrándolos arriba o abajo, o haciéndolos hijos o padres.
Inspección	Se usa para ver y editar propiedades de objeto y también preferencias y otros ajustes dentro de Unity
Herramienta	Son cinco controles básicos. Cada uno se relaciona con diferentes partes del Editor. Se trata de un acceso rápido.
Juego	Es el renderizado que me genera la cámara de la escena. Es la representación del modelo ya finalizado

Figura 17. Estructura organizativa en unity 5

- Game Object →padre→hijo: Unity usa este concepto para hacer que cualquier Game Object sea el hijo deseado al padre. Un hijo heredará el movimiento y rotación de su padre.
 - Script: Es un lenguaje de programación que ejecuta diversas funciones en el interior de un programa informático. En mi caso hemos utilizado #C Sharp
 - FPS: También conocido controlador del personaje en primera persona. Se define como un objeto que permitirá manejar un personaje en la pantalla, es decir, posibilita la visualización del entorno desde la visión del personaje
4. Materiales: Debido a la pérdida de materiales Unity 5 reconoce todo el elemento como un bloque en blanco, por tanto, para aplicarle realismo al edificio habrá que:
 - A. Crear materiales: En la ventana de jerarquía hacemos clic derecho y elegimos la opción de materiales, a continuación, en la ventana de inspector vamos a añadirle algunas propiedades como el color.
 - B. Colocar texturas: Dentro de la misma pestaña anterior si bajamos a la opción que dice albedo podemos también añadirle una textura.
 - C. Asignar al objeto: Una vez definido los dos puntos anteriores la asignación del material al objeto es muy simple, basta con arrastrar el material hasta el objeto deseado.
 5. Lenguaje de programación: Este es uno de los puntos más importantes del proyecto ya que esta herramienta nos permitirá movernos alrededor de la escena y además dependiendo de la orden que se le proponga nos proporcionará una información determinada o una orden, como, por ejemplo, la apertura automática de puertas o el etiquetado de elementos.

E. Ficha técnica: Características específicas de Revit

Características específicas de 3ds Max				
Características	Comprobación previa	Programa	Tipo	Ruta
Escala	1:100			
Unidades	mm			
Vincular		Unity 5		
Formato			FBX	Escritorio
Escala				
Propósito	Poder trabajar con ambos modelos para contrastar la información de la ficha técnica: Caso de uso 2			
Planteamiento	Preparar el modelo de Revit para abrir en Unity 5			
Objetivo	Visualizar, analizar y recorrer a través del edificio propuesto en el caso de uso 2			
Resultado	Conseguido			

Figura 18. Estructura organizativa en Revit

Como información complementaria al caso de uso 1 propuesto, también se ha adjuntado dos modelos de prueba vinculados al diseño constructivo y modelado, estos van a ir referidos al etiquetado de elementos y la apertura automática de puertas.

Aunque en el capítulo 4 pasaremos a explicar la unión de los dos casos de uso, como un proyecto único. A continuación, evaluaremos los modelos por separado debido a la complejidad de ejecución.

4.1. Caso de estudio 1: Estructura + Arquitectura



Figura 19. Orden cronológico de trabajo. Caso 1

El modelo se inicia con:

1. AutoCAD:

- Comprobaremos
 - A. Planos de las distintas plantas del edificio
 - B. Escala del proyecto
- Objetivo: Visualizar sus planos en Unity 5 [se puede ver en el capítulo 4]
- Resultados: Conseguido. Su valoración será analizada en el capítulo 4.

2. 3ds Max: Aquí es donde se inicia el trabajo, propiamente dicho para ello vamos a plantear un esquema organizativo donde se tendrá en cuenta los siguientes puntos:

- Planteamiento previo
 - ✓ Evaluar que elementos son necesarios para la escena y cuales no
- Desarrollo
 - ✓ Eliminar todos los objetos que son considerados como secundarios (luces y cámaras)
- Conclusión
 - ✓ Como se dijo al principio de este capítulo, lo que se busca es poder abrir el fichero de 3ds Max en Unity 5, para ello primero debemos de evaluar y

buscar la forma de optimizar el modelo para que cuando se lleve a Unity este tenga el menor peso posible, ya que de lo contrario tanto el recorrido virtual como la hipotética visualización a través de un dispositivo móvil se vería gravemente ralentizada.

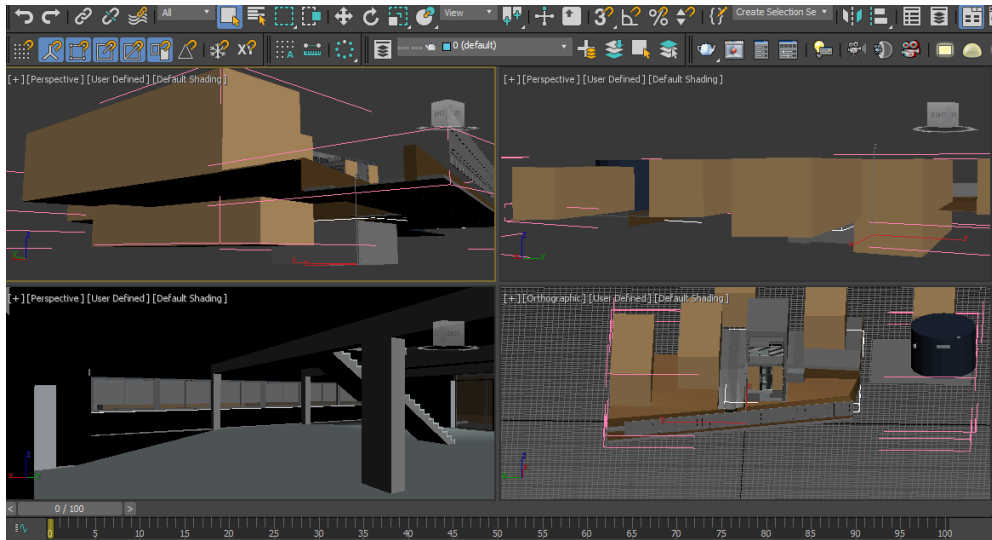


Figura 20. Perspectiva del modelo final en 3ds Max

- Resultado: de la evaluación de la compatibilidad de formatos de la cual hemos hablado al inicio del capítulo, sabemos que el formato compatible es FBX, por tanto, vamos a proceder a guardar el archivo, donde en un inicio se dejara guardado el modelo en el escritorio o en la carpeta donde se quiera.

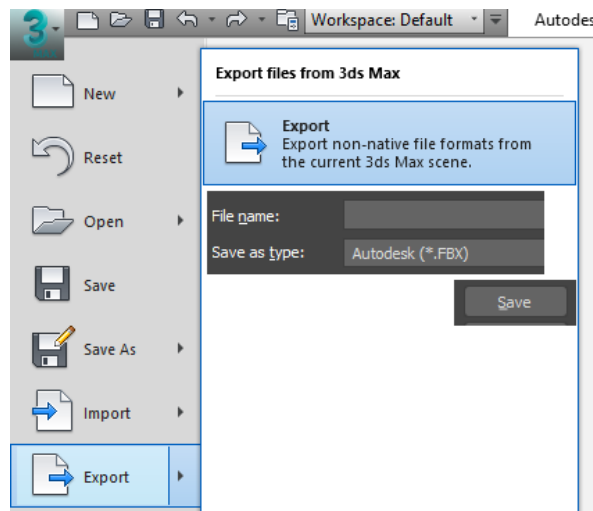


Figura 21. Propiedades de formato de exportación en 3ds Max

3. Unity 5: Es donde se ejecutado todo nuestro trabajo, para explicar de la mejor manera posible me he creado cuatro apartados que son:

- Planteamiento
 1. Creamos un proyecto nuevo llamado Arquitectura
 2. Visualizamos la diferentes sub-ventas y localizamos la ruta de la carpeta de Asset y guardamos el modelo definido en el punto 2
 3. Arrastramos el modelo hacia la venta de juego
 4. Comprobación la escala Es recomendable pero no obligatorio mover el origen al (0,0,0).

- Desarrollo
 1. Creamos varias sub-carpetas: Con el objetivo de mantener un orden en toda la escena y facilitar así el trabajo realizado.
 2. A todo el edificio le colocaremos un mallado de colisión, esto nos ayudara a que cuando ejecutemos el comando de visualización el personaje que crearemos más adelante no caiga al infinito
 3. Asignaremos a cada objeto su material correspondiente

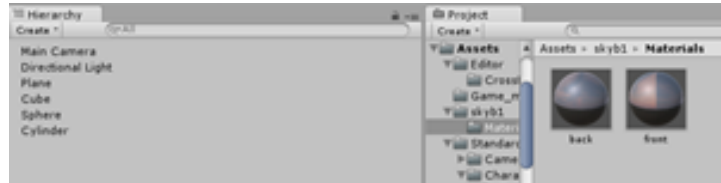


Figura 22. Propiedades de los materiales en Unity 5



Figura 23. Asignación de material al objeto ladrillo en fachada en Unity 5

4. Creación de ambiente y luces: Para ello se han creado varios puntos de luz, su creación es relativamente sencilla, basta con dibujar cualquier figura geométrica (figura 24), y en las opciones de luz escoger punto de luz (figura 25) este nos permitirá visualizar el edificio en un ambiente nocturno como es nuestro caso

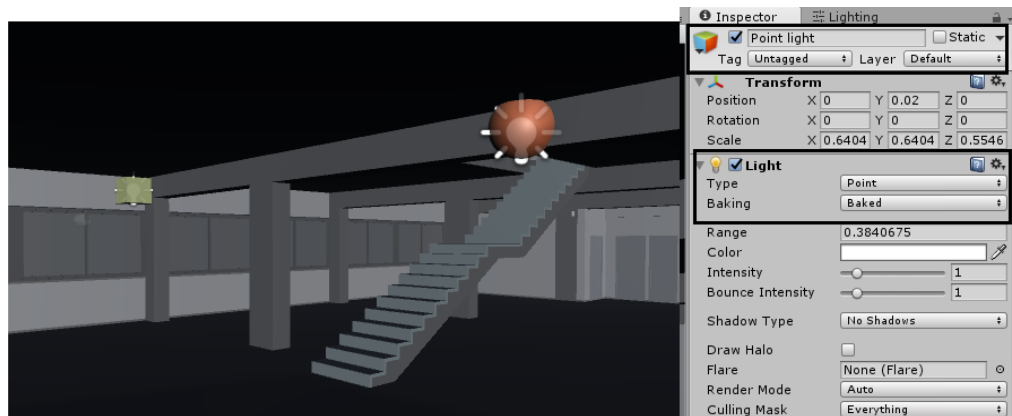


Figura 24. Propiedades características de luz



Figura 25. Propiedades características de las componentes

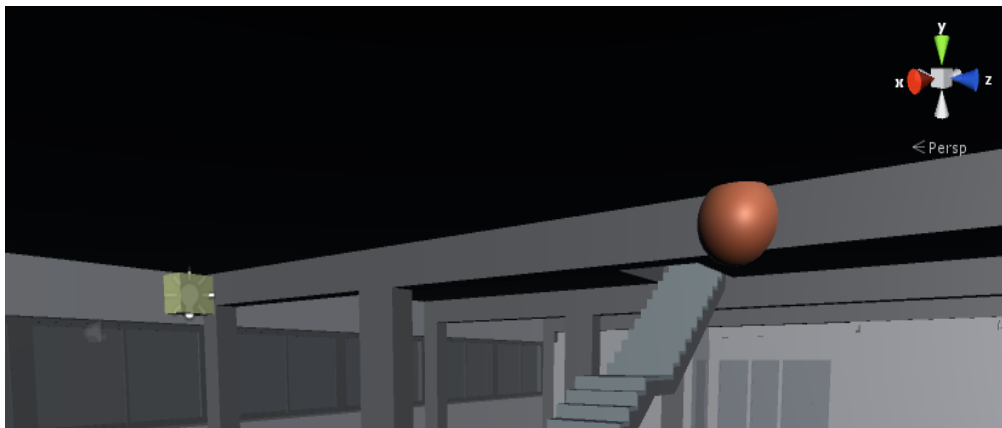


Figura 26. Vista interior de la planta baja del Ed. B0. CIMNE. UPC Campus Nord

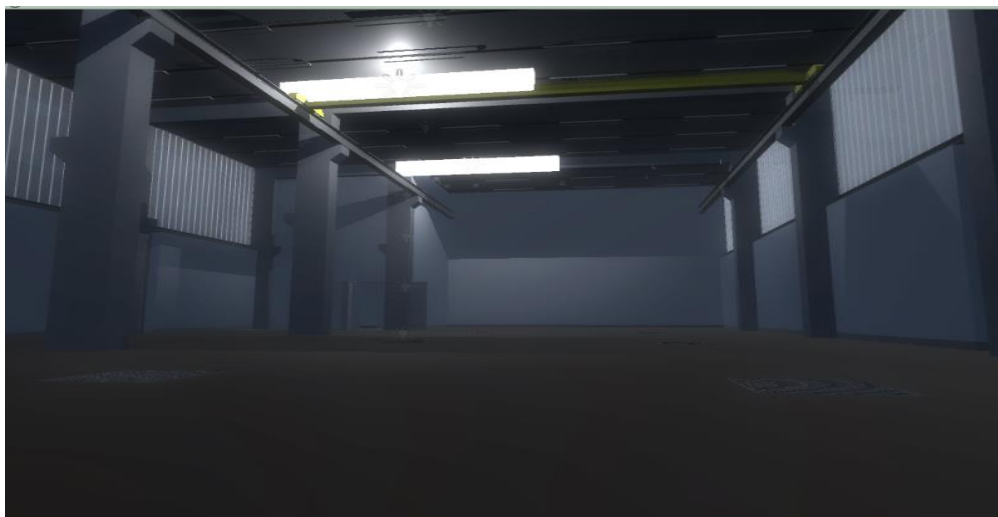


Figura 27. Visualización de la planta sótano



Figura 28. Visualización exterior del Campus Nord UPC en Unity 5

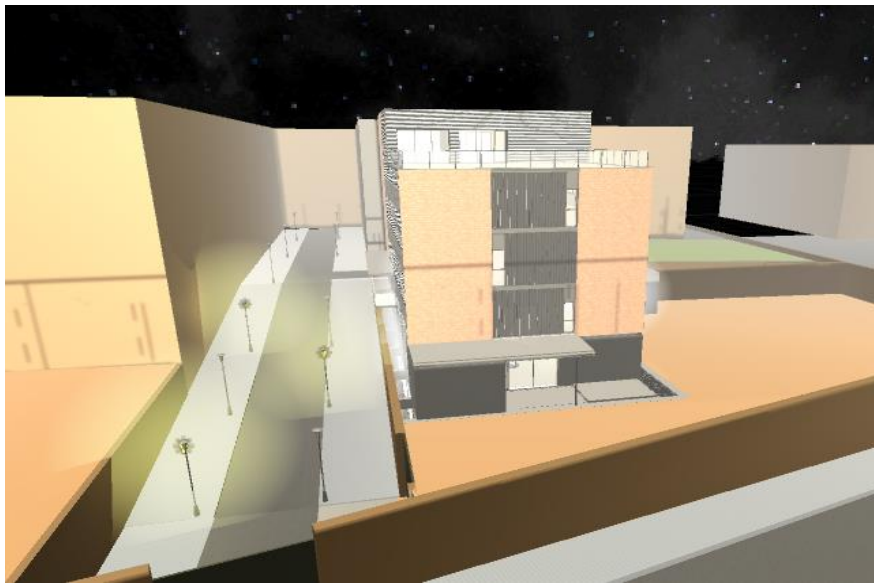


Figura 29. Visualización exterior del Ed. B0. CIMNE. Campus Nord con Unity 3d

5. Creación de script

5.1. Puertas: Para su visualización se pueden hacer de dos maneras

- A. Estática es decir que está abierta o cerrada en todo su recorrido
- B. Automática es decir conforme se avance en el recorrido se abrirá y se cerrará de manera automática.

En este proyecto se han optado por el diseño de las dos, donde como se puede ver en la maqueta digital del proyecto las puertas que conectan las plantas se encuentra abierta permanente.

Para la creación de puertas automáticas, es necesario un trabajo previo ya que primero hay que crear scripts, estos darán ordenes al modelo consiguiendo así su apertura. El lenguaje y visualizador que se ha utilizado para su ejecución son, Visual Studio 2016 en un lenguaje en Java script.

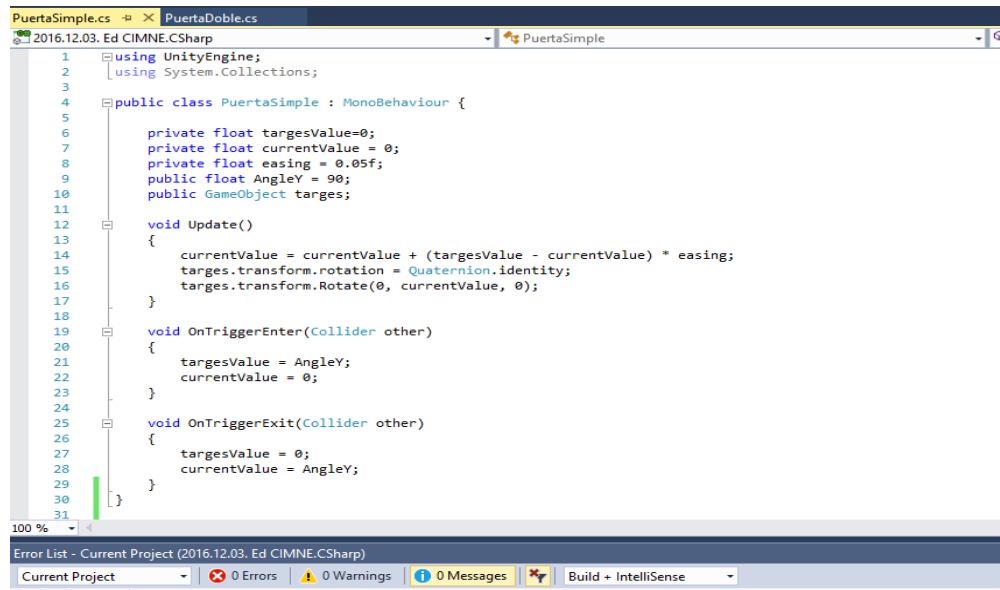


Figura 30. Ejemplo de código de programación en C Sharp para puerta simple en Visual Studio

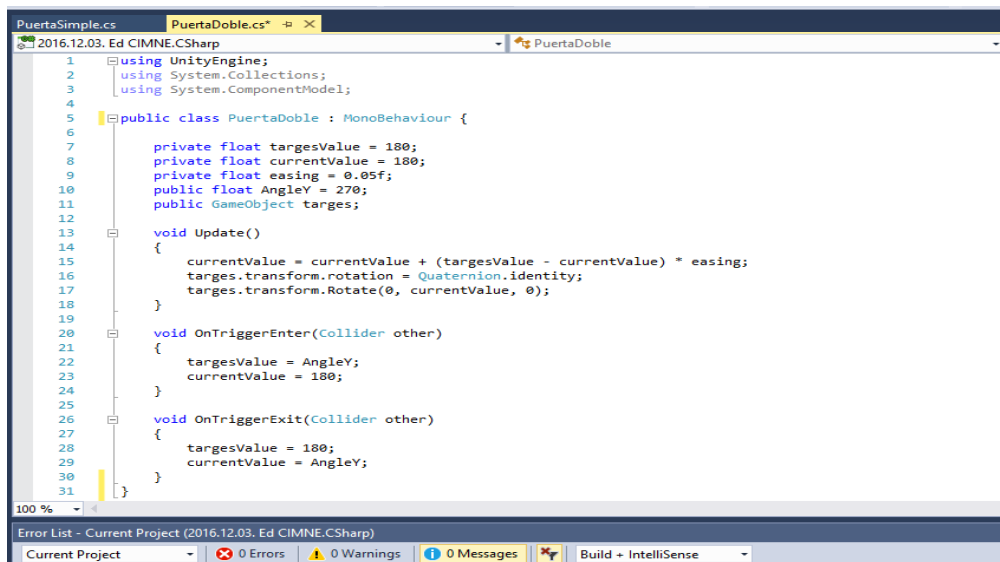


Figura 31. Ejemplo de código de programación en C Sharp para puerta doble en Visual Studio

Como se puede ver en la figura 30 y 31 creamos dos códigos diferentes uno va referido a una puerta de una sola hoja con apertura a 90 grados mientras que la segunda está pensada para una puerta doble con un ángulo de apertura de 180 grados.

Para entender de manera más clara se creó un modelo auxiliar en este aplicaremos todas las ordenes y luego las aplicaremos al proyecto.

Para ello vamos a partir de una idea sencilla y es con la creación de un cuadrado que hará de bisagra, girara en torno al eje Y un rectángulo que será la puerta, se crearan a través de un cuadrado.

Colocamos la puerta sobre la bisagra para comprobar que gira alrededor de su propio eje, aquí se pueden dar dos casos:

A. Bisagra como padre y puerta como hijo

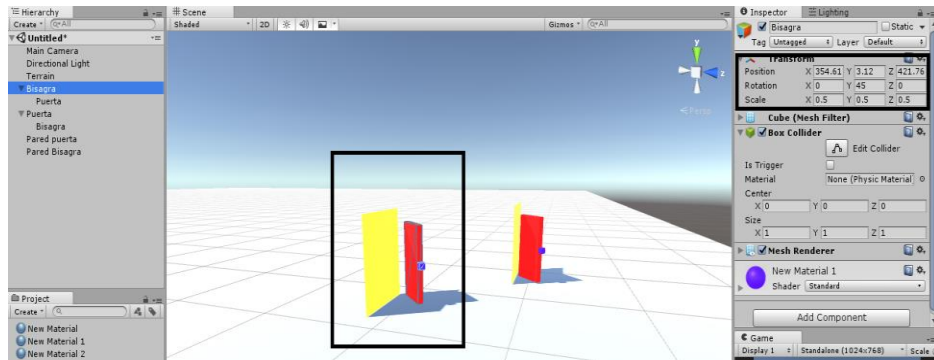


Figura 32. Bisagra como padre en Unity 5

B. Puerta como padre bisagra como hijo

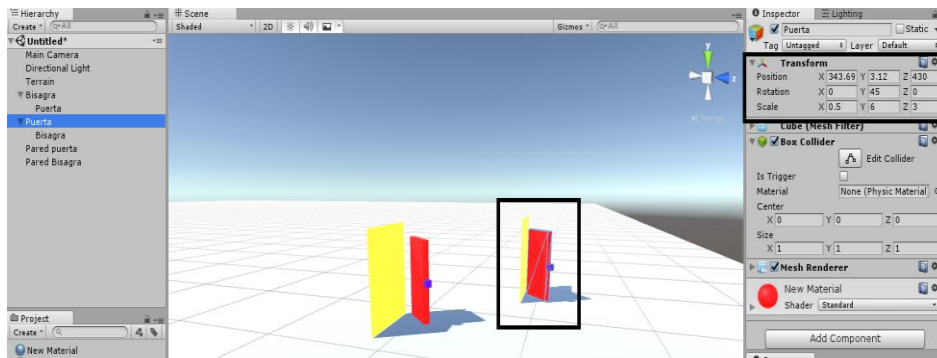


Figura 33. Puerta como padre en Unity 5

Esto dependerá fundamentalmente del sentido en el que se quiere que gire la puerta y de las características del mismo, en la figura 32 y 33 se han mostrado los dos posibles casos.

En ambos casos se selecciona la bisagra y en la ventana de inspector vamos a desactivar la opción de visualización de mallado ya que no queremos que el cuadrado creado para la bisagra se visualice en la escena, también está la opción de visualizar, esto dependerá del criterio y gusto de la persona que lo diseña, en mi caso opte por ocultarle.

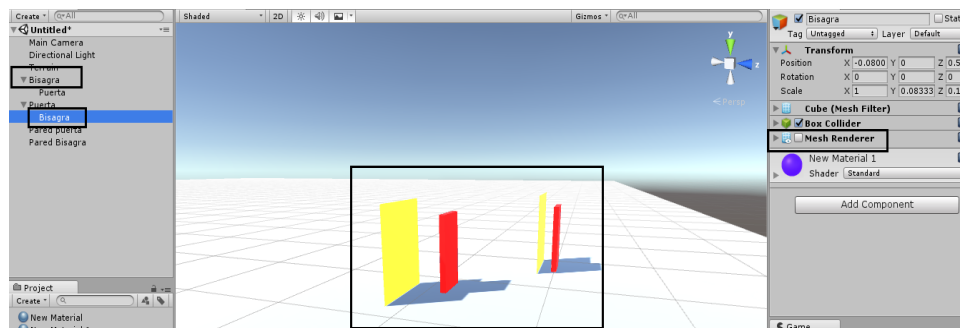


Figura 34. Propiedades de desactivación de mallado en Unity 5

Hecho esto, se ha de crear otro subo, este nos permitirá que detecte el PFS cuando se acerque a la puerta esta se abra de manera automática, es decir, nos va

a proporcionar el límite de detección de la puerta. El ancho puede variar todo dependerá siempre de la situación de cada caso.

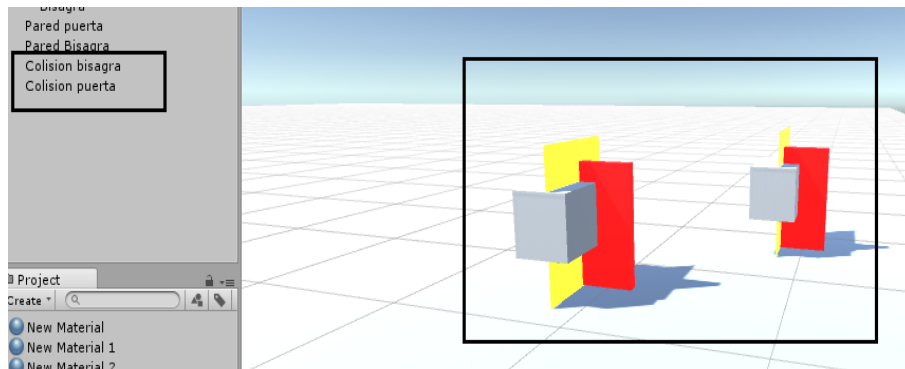


Figura 35. Creación de las colisiones en Unity 5

Se ha de crear además un Game Object, nos ayudara a ordenar todos los elementos que presenta el escenario puerta figura 36. Como se puede observar se ha renombrado como GO. Bisagra dentro de él está el padre y su hijo además del script de apertura de puerta.

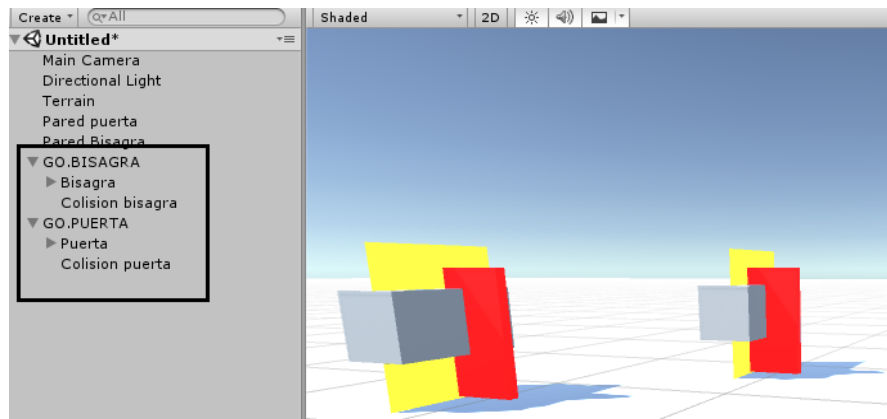


Figura 36. Orden jerarquía de trabajo en Unity 5

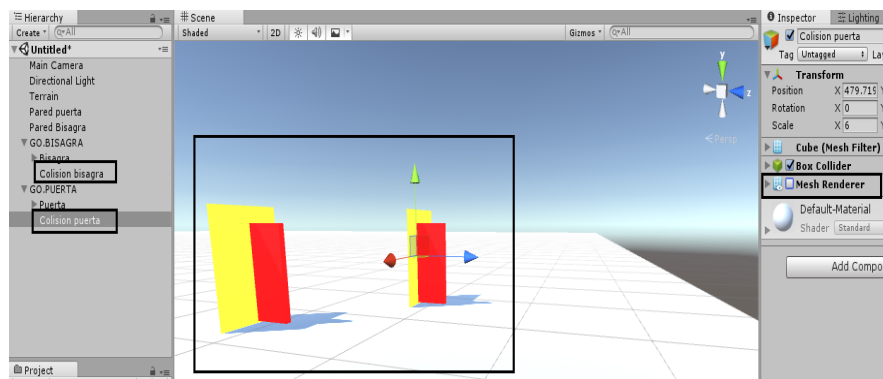


Figura 37. Propiedades específicas de mallado en Unity 5

Si nos fijamos en las propiedades de la ventana de inspector de la figura 37 y 38 vemos que para que nos funcione es necesario activar la función is trigger. Esta se ejecuta cuando un objeto entra dentro de otro.

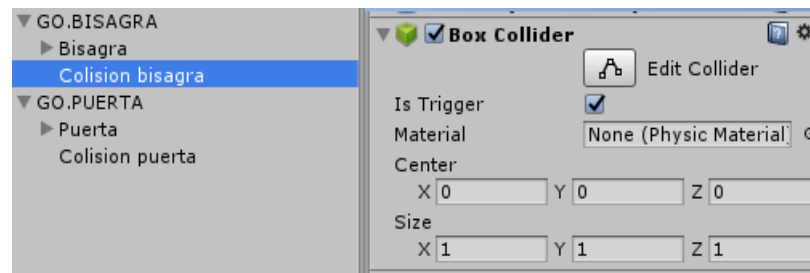


Figura 38. Propiedades específicas de colisión en Unity 5

A continuación, agregamos el script creado

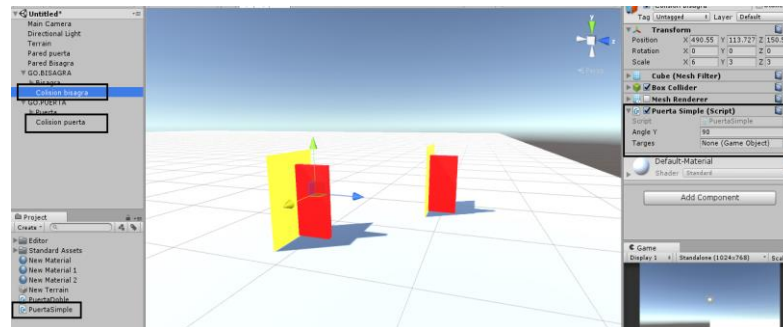


Figura 39. Propiedades específicas de script en Unity 5

Previa a la comprobación en la misma ventana del script si nos vamos a sus propiedades tenemos una opción que se denomina tarjes. Esta nos dice que tenemos que colocar el elemento con el cual se quiere efectuar y detectar la colisión figura 40

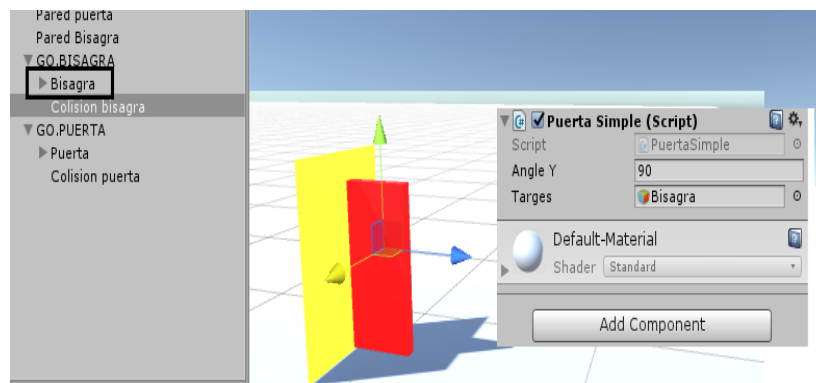


Figura 40. Función tarjes. Caso cuando bisagra actúa como padre en Unity 5

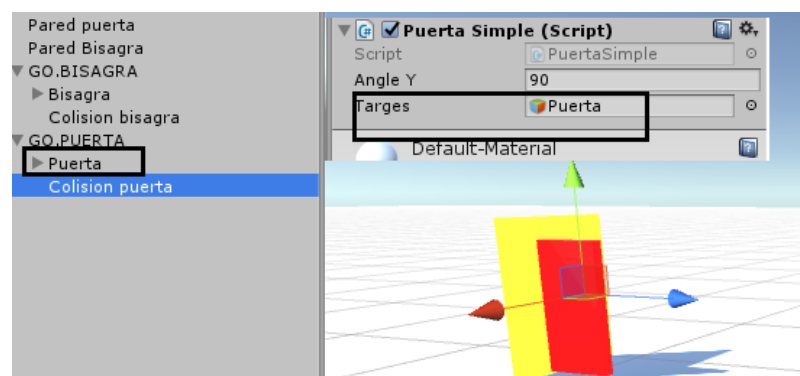


Figura 41. Función tarjes. Caso cuando la puerta actúa como padre en Unity 5

Por ultimo vamos a proceder a comprobar si lo creado es correcto, para ello clicamos en PLAY. Como se puede ver en la figura 42 funciona.

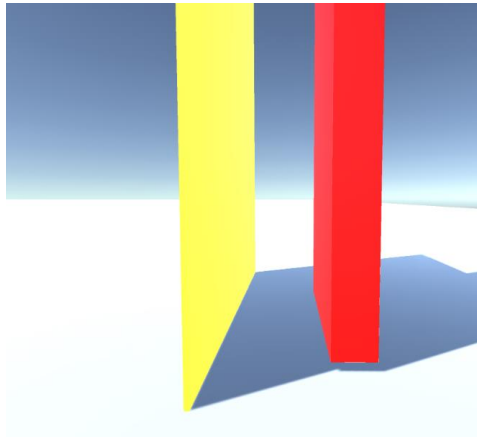


Figura 42. Visualización en Unity 5

Si aplicamos lo anterior a nuestro proyecto en la figura 43 y 44 vemos que funciona

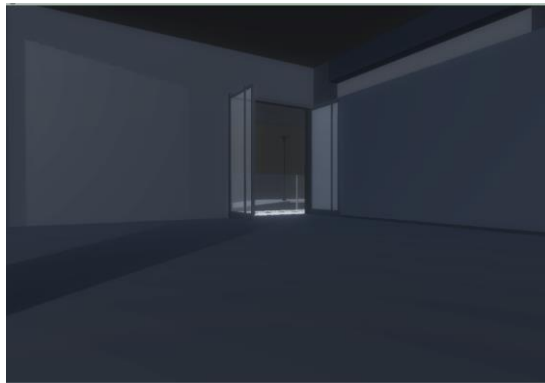


Figura 43. Visualización puerta lateral P1 en Unity 5

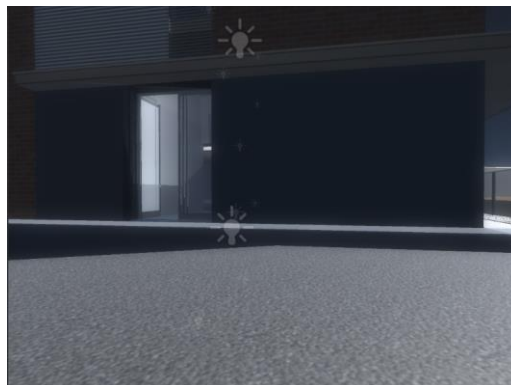
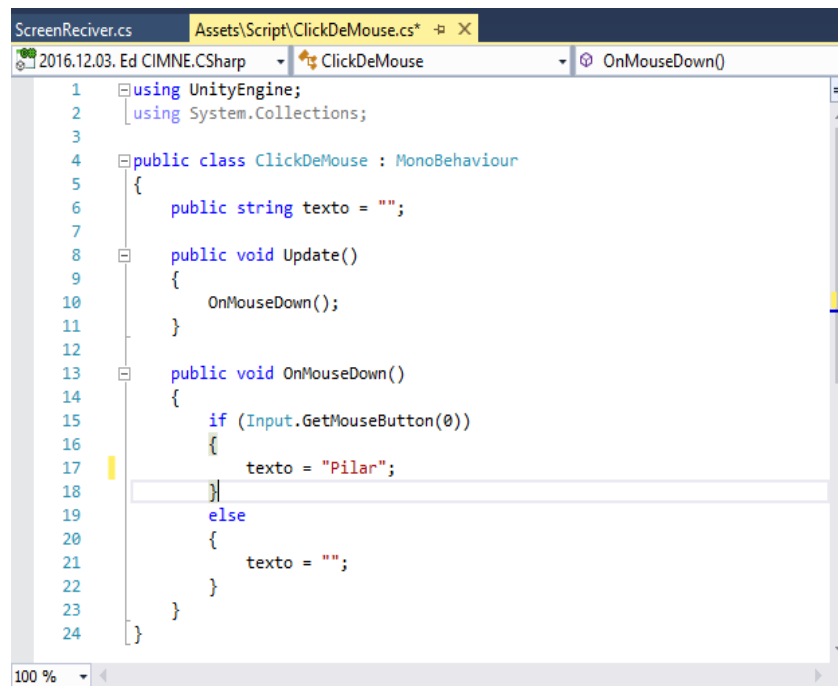


Figura 44. Visualización puerta trasera P1 en Unity 5

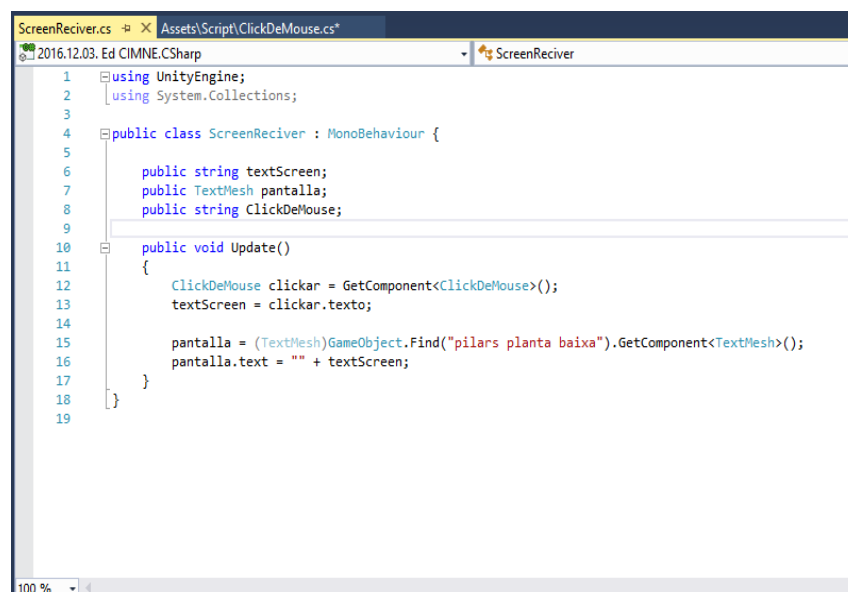
5.2.Etiquetado de objetos: Al igual que el anterior se han de crear scripts para que conforme avance el personaje por la escena este sea capaz de detectar y automáticamente generar el mensaje de información sobre algún elemento del edificio.

En la figura 45 y 46 hemos creado dos scripts uno ira referido al persona y otro al objeto donde quiero colocar el mensaje

The image shows a Visual Studio code editor window with the file 'Assets/Script/ClickDeMouse.cs' open. The script is a C# class named 'ClickDeMouse' that inherits from 'MonoBehaviour'. It contains a public string 'texto' initialized to an empty string. The 'Update()' method calls 'OnMouseDown()'. The 'OnMouseDown()' method checks if the left mouse button (0) is pressed. If it is, it sets 'texto' to 'Pilar'; otherwise, it sets 'texto' back to an empty string. The code is as follows:

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class ClickDeMouse : MonoBehaviour
5 {
6     public string texto = "";
7
8     public void Update()
9     {
10         OnMouseDown();
11     }
12
13     public void OnMouseDown()
14     {
15         if (Input.GetMouseButton(0))
16         {
17             texto = "Pilar";
18         }
19         else
20         {
21             texto = "";
22         }
23     }
24 }
```

Figura 45. Ejemplo de código de programación en C Sharp para ClickDeMouse en Visual Studio

The image shows a Visual Studio code editor window with the file 'Assets/Script/ScreenReciver.cs' open. The script is a C# class named 'ScreenReciver' that inherits from 'MonoBehaviour'. It contains three public variables: 'textScreen', 'pantalla' (of type TextMesh), and 'ClickDeMouse'. The 'Update()' method finds the 'ClickDeMouse' component and updates 'textScreen' with its 'texto' property. It also finds a 'TextMesh' component for the object 'pilars planta baixa' and updates its 'text' property with 'textScreen'. The code is as follows:

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class ScreenReciver : MonoBehaviour {
5
6     public string textScreen;
7     public TextMesh pantalla;
8     public string ClickDeMouse;
9
10    public void Update()
11    {
12        ClickDeMouse clickar = GetComponent<ClickDeMouse>();
13        textScreen = clickar.texto;
14
15        pantalla = (TextMesh)GameObject.Find("pilars planta baixa").GetComponent<TextMesh>();
16        pantalla.text = "" + textScreen;
17    }
18 }
19
```

Figura 46. Ejemplo de código de programación en C Sharp para ScreenReceiver En Visual Studio

Para iniciar su creación, partimos por ejemplo con un cubo para que detecte la colisión cuando FPS se acerque al límite que se marca y a continuación nos aporte la información que queremos.

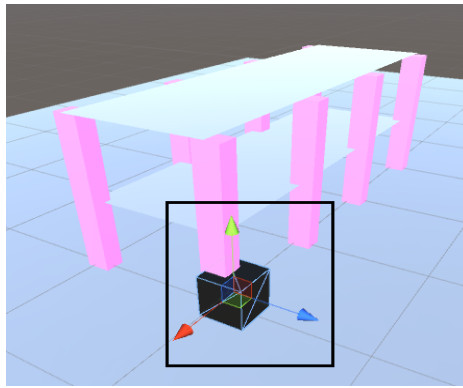


Figura 47. Creación de la colisión en Unity 5

este cubo como se ve en la figura 47 se coloca cerca del elemento al que queremos nombrar

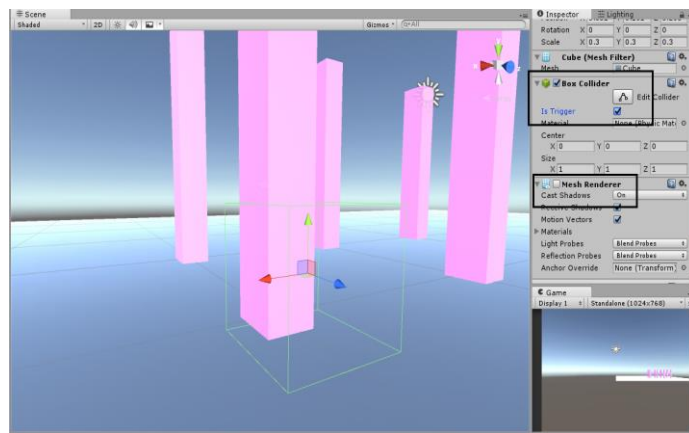


Figura 48. Propiedades de la caja de colisión en Unity 5

A continuación, creamos otro script que nos aportara la información que se quiere mostrar en la pantalla cuando el personaje se acerque al objeto

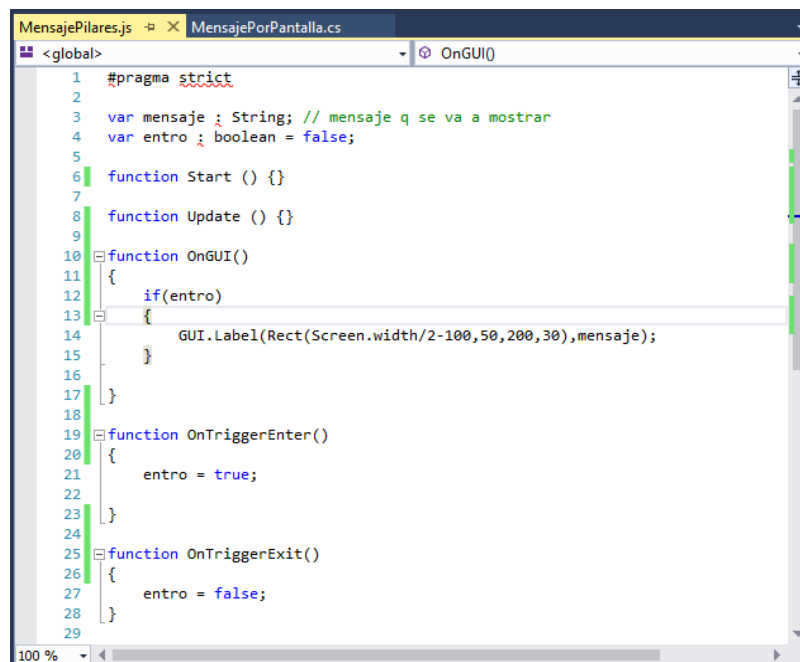


Figura 49. Ejemplo de código de programación en C Sharp para etiquetado en Visual Studio

En la figura 49 nos muestra el script que dará la orden para la creación del mensaje, le colocare a (100,50,200), estas medidas las he sabado en la posición del pilar. Cabe destaca que su posición puede variar. Cundo está terminado el script se le asigna al cubo creado para ello solo basta con arrastrar y listo (figura 50)

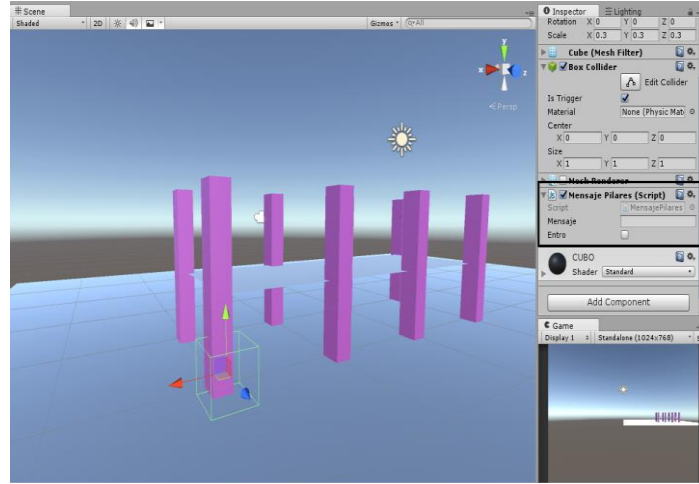


Figura 50. Colocación del elemento colisión en pilar extremo en Unity 5

Si se mira con más detalle se puede ver que una de las componentes del script va referenciada a mensaje, en esta vamos a escribir la información que se quiera, por ejemplo, para el caso que se presenta en el caso de prueba es llamar pilar extremo

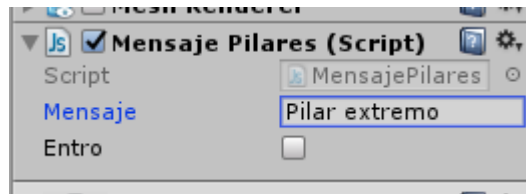


Figura 51. Propiedades de mensaje en Unity 5

Por ultimo nos queda probar y ver si funciona. Como se puede ver en la figura 52 conforme el personaje se ha ido acercando el mensaje a saltado. Por tanto, funciona

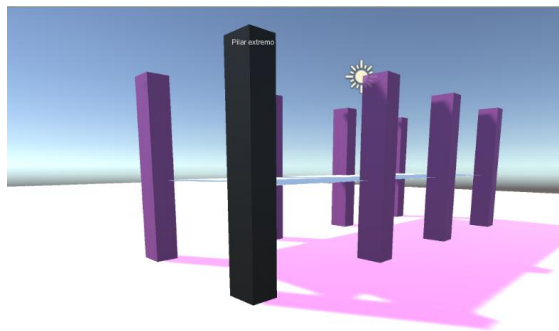


Figura 52. Resultado final de etiqueta en Unity 5

Lo creado en el modelo de prueba se ha de llevar a cabo en nuestro modelo donde el resultado que se ha obtenido es el siguiente.

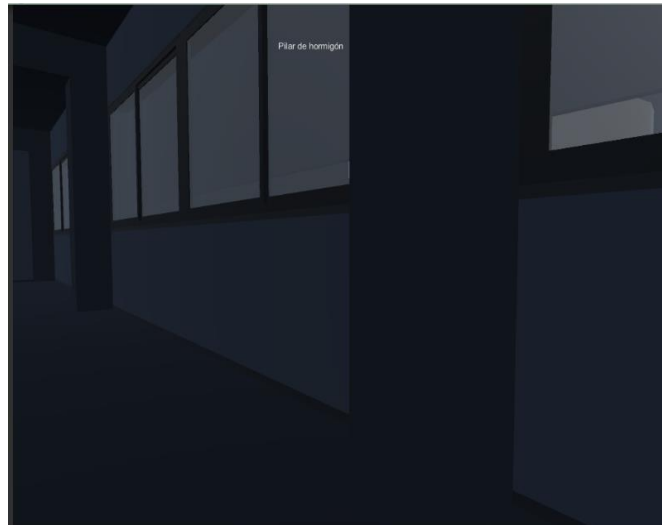


Figura 53. Mensaje. Etiquetado pilares en Unity 5

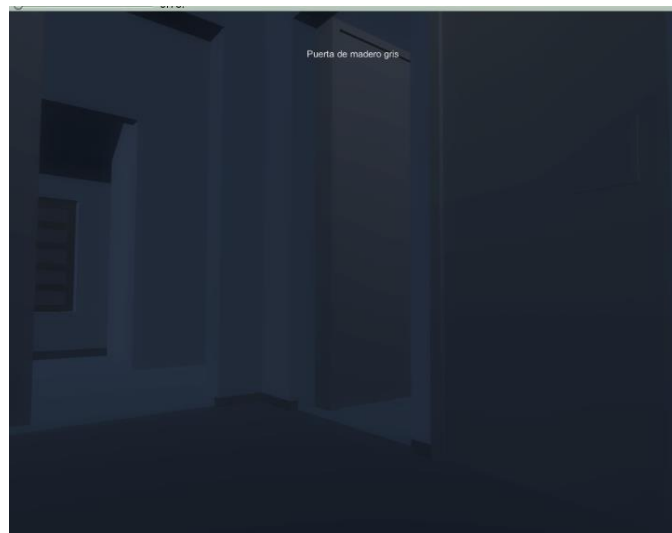


Figura 54. Mensaje. Etiquetado puertas en Unity 5

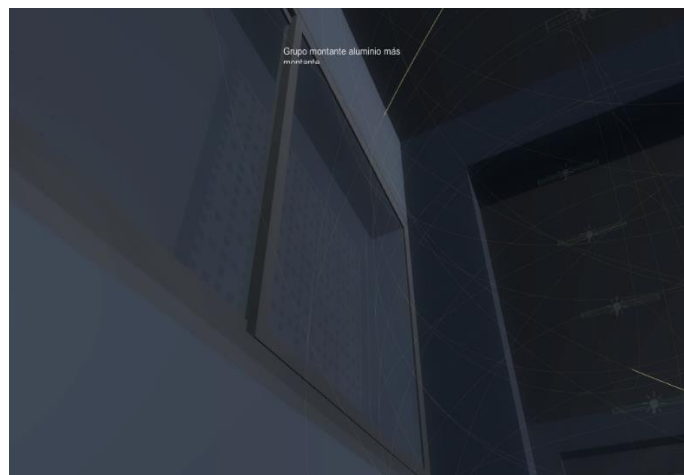


Figura 55. Mensaje. Etiquetado ventanas en Unity 5

- 5.3. FPS: Una de las ultimas cosas que nos colocar es el personaje que se moverá por la escena. Su colocación es sencilla solo basta con ir a una de las carpetas de Asset denominada controladores de carácter y elegir controlador en primera persona, solo basta arrastrar este a la escena y tenemos creado nuestro FPS. No es necesario crear script ya que Unity 5 por defecto tiene una carpeta predeterminada con algunas funciones y una de ellas son las características del personaje.
- Conclusión: De este primer estudio podemos sacar varias ideas.
 1. A través de la compatibilidad de formatos es posible trabajar con diferentes programas. En mi caso 3ds Max – Unity 5
 2. Mediante la optimización de la estructura además hemos logrado que el modelo tenga un peso optimo, donde la visualización y recorrido tienen una buena fluidez.
 3. Añadiendo características y propiedades tanto a la escena como a los objetos podemos mostrar de manera detallada el funcionamiento del edificio y así poder comparar y estudiar todos los puntos marcados en la figura 13.
 - Resultado: Como se puede observar en la figura 56 y una vez finalizada la colocación de todos sus componentes funciona.

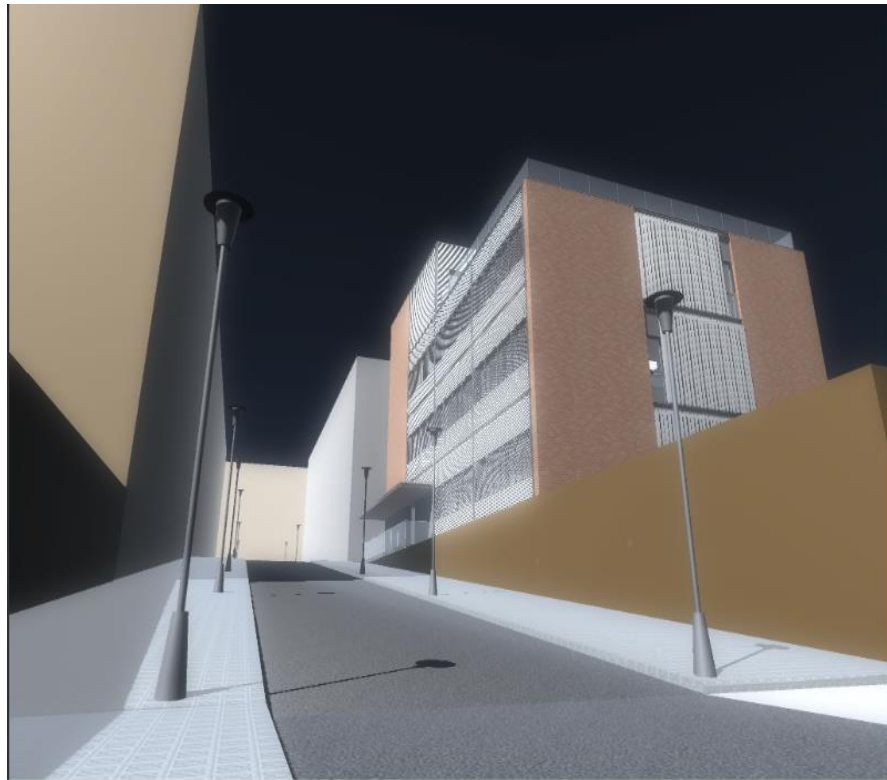


Figura 56. Vista exterior. Ed.B0. CIMNE. Campus Nord para Unity 5

4.2. Modelo gráfico 2: Instalaciones



Figura 57. Orden cronológico de trabajo. Caso

1. Modelo revit: Partimos de un modelo creado en la plantilla MEP (viene de Mechanical, Electrical y Plumbing es una de las tres disciplinas por las que está compuesta Revit).
 - A. Planteamiento: En la figura 58 izquierda se puede ver el edificio tal cual esta creado en Revit, en la imagen de la derecha lo que se ha hecho es dejar todas las instalaciones vistas para ver qué tipo de instalación tiene y así poder comparar en Unity 5

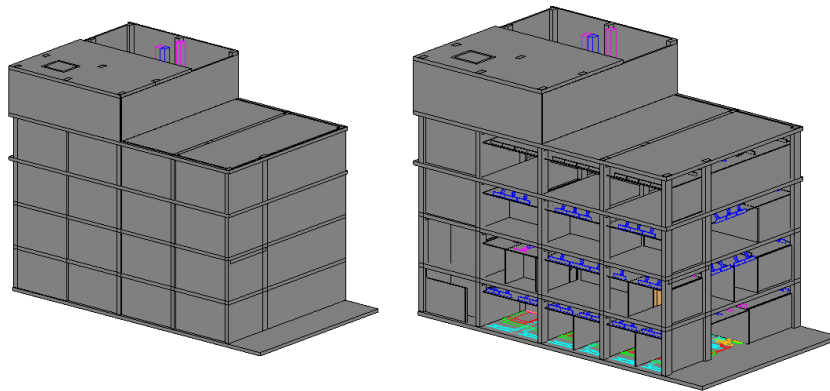


Figura 58. Edificio modelado en Revit

- B. Ejecución: Previa a la importación a Unity 5, tenemos que comprobar
 1. Comprobar la escala previa a la importación es muy importante ya que de no concordar las dimensiones cuando se abra el archivo en Unity 5 pueden llegar a ser exageradas



Figura 59. Comprobación: Escala de trabajo en Revit

2. Comprobar si tenemos vinculada alguna plantilla de AutoCAD t si es así hay que eliminar ya que a la hora de importar a Unity 5 está lo único que nos aportará será su peso, no aporta valor informativo por tanto se considera un elemento secundario el cual se puede obvia.

Manage Links						
Revit IFC CAD Formats DWG Markups Point Clouds						
Link Name	Status	Positions	Size	Saved Path	Path Type	
Base P2.dwg	Not Found	<input type="checkbox"/>	132.0 K	Base P2.dwg	Relative	
Base P3.dwg	Not Found	<input type="checkbox"/>	117.8 K	Base P3.dwg	Relative	
Base P4.dwg	Not Found	<input type="checkbox"/>	93.0 KB	Base P4.dwg	Relative	

Figura 60. Comprobación: Coordenadas del proyecto en Revit

- C. Resultado: Una vez comprobado los dos pasos anteriores se procederá a guardar y como en el caso 1 el formato en el que se guarda debe ser compatible con Unity 5. Por tanto, vuelvo a recurrir al formato FBX

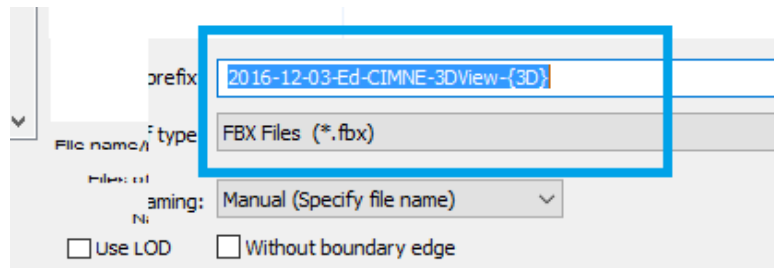


Figura 61. Proceso de formato importación Revit – Unity 5

2. Modelo Unity:

A. Planteamiento

1. Creamos un nuevo archivo en Unity 5, y lo llamamos 2016.12.03. Ed CIMNE_Instalación
2. Al abrir el archivo igual que en el caso 1
 - 2.1. Primero guardamos en la carpeta de Asset el modelo de Revit
 - 2.2. Miramos la estructura organizativa, es decir las sub-ventanas.

Figura 17.



Figura 62. Ruta del proyecto creada en Unity 5

3. Una vez hecho lo anterior desde la carpeta de Asset vamos a arrastrar nuestro proyecto hasta la ventana de jugador
 - 3.1. Comprobamos la escala 1:100
 - 3.2. Orientamos todo al centro de origen, es decir, (0,0,0), este paso es opcional yo lo realizo para facilitar mi trabajo cuando cree sobre todo los scripts ya que por ejemplo si quiero que empiece en la planta baja el recorrido partiré de origen de coordenadas (0,0,0) y no tendré que ir probando hasta dar con su punto exacto en el que el programa me coloco en un inicio.

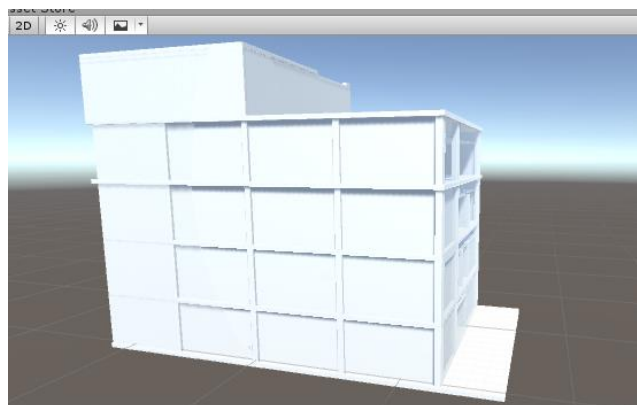


Figura 63. Vista inicial del edificio en Unity 5

- 3.3. Al igual que hicimos en el caso de uso 1, vamos a colocar materiales para ello utilizaremos la misma carpeta del modelo anterior donde como resultado nos quedara tal que:

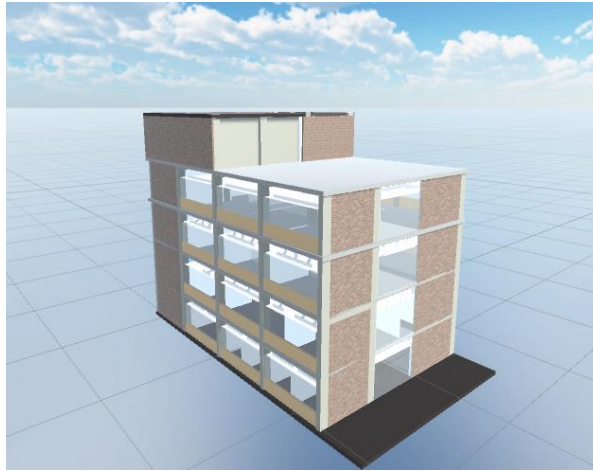
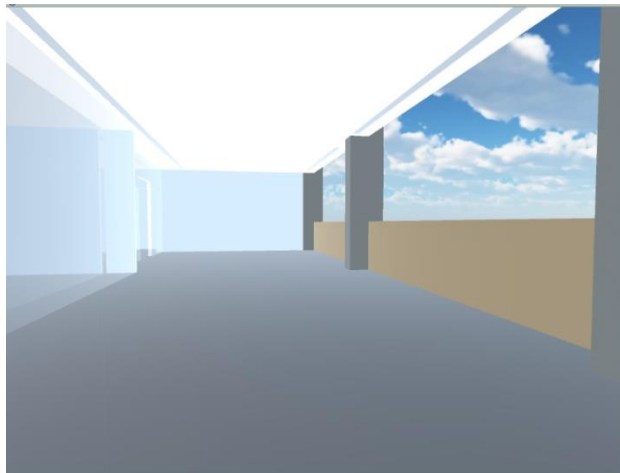


Figura 64. Vista final del edificio en Unity 5

- 3.4.A diferencia del otro modelo, en este nos vamos a centrar en una planta y será la 2º planta del ed. B0. CIMNE. UPC Campus Nord. Nos centramos en esta porque en el capítulo 5 vamos a comparar los dos modelos. Por tanto, este será el punto donde debo colocar mi controlador en primera persona o FPS

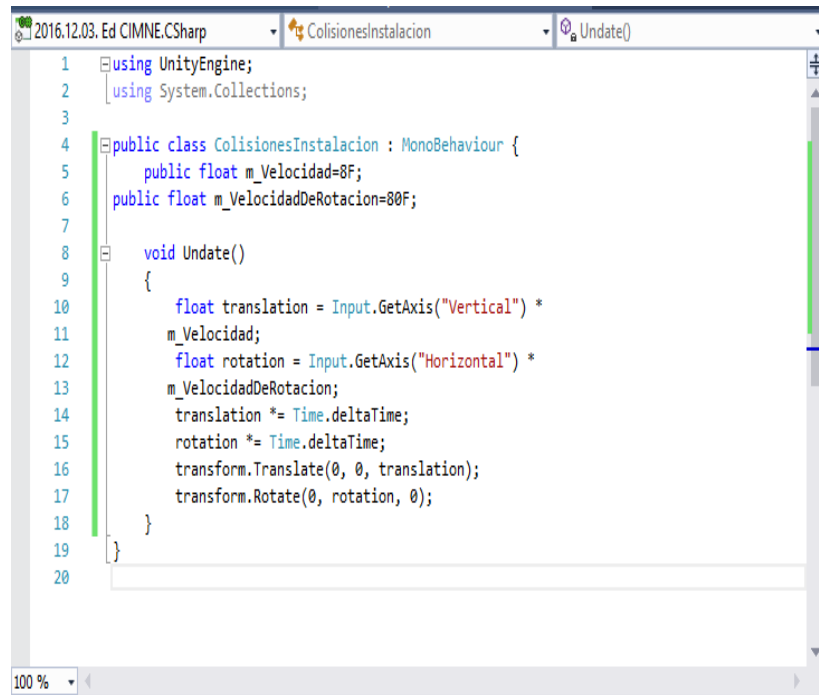


**Figura 65. Posición del FPS en la 2 planta del
Ed. B0. CIMNE. UPC Campus Nord en Unity 5**

B. Desarrollo

1. Creación de script: Como se ha dicho en el modelo anterior, si queremos dar realismo a la escena, poder recorrer en su interior o evaluar los elementos de estudio planteados en la figura 14, es necesario la creación de script:

- 1.1. Detección de colisión: Para poder tener una visualización exclusiva solo de las instalaciones hay que crear un script de orden bloque elementos. A esta script le llamaremos “ColisionesInstalacion”, lo que se consigue con esto es poder movernos dentro de la escena y que conforme el personaje avance los objetos vayan desapareciendo uno a uno, dejando solo visible la instalación de la planta de estudio.



```
1 using UnityEngine;
2 using System.Collections;
3
4 public class ColisionesInstalacion : MonoBehaviour {
5     public float m_Velocidad=8F;
6     public float m_VelocidadDeRotacion=80F;
7
8     void Update()
9     {
10         float translation = Input.GetAxis("Vertical") *
11             m_Velocidad;
12         float rotation = Input.GetAxis("Horizontal") *
13             m_VelocidadDeRotacion;
14         translation *= Time.deltaTime;
15         rotation *= Time.deltaTime;
16         transform.Translate(0, 0, translation);
17         transform.Rotate(0, rotation, 0);
18     }
19 }
20
```

Figura 66. Ejemplo de código de programación en C Sharp para colisión en Visual Studio para Unity 5

Se ha de agregar también un RayCast, a grosso modo un rayo invisible pero visible en la escena. Nos sirve para detectar el objeto en la escena la cual se encuentra a una distancia determinada

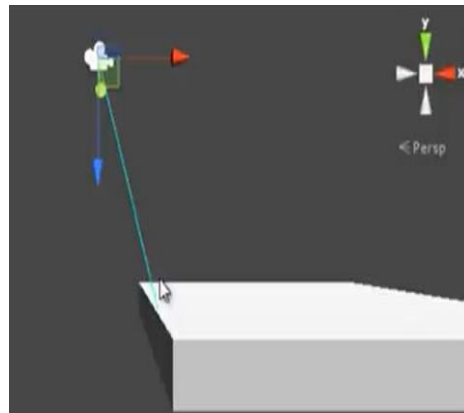


Figura 67. Rayo de detección de la escena en Unity 5

- 1.2. Rigidbody: Con esta script permitirá al objeto en cuestión actuar bajo el control de la física, es decir vamos a conseguir que los objetos se muevan en una manera realista. Hay que decir que cualquier GameObject debe contener un Rigidbody porque si no, no podrá estar influenciado por la gravedad. Por tanto, si queremos que el script anterior funcione de manera correcta y actúe o interactúe con otros objetos es necesario su creación.

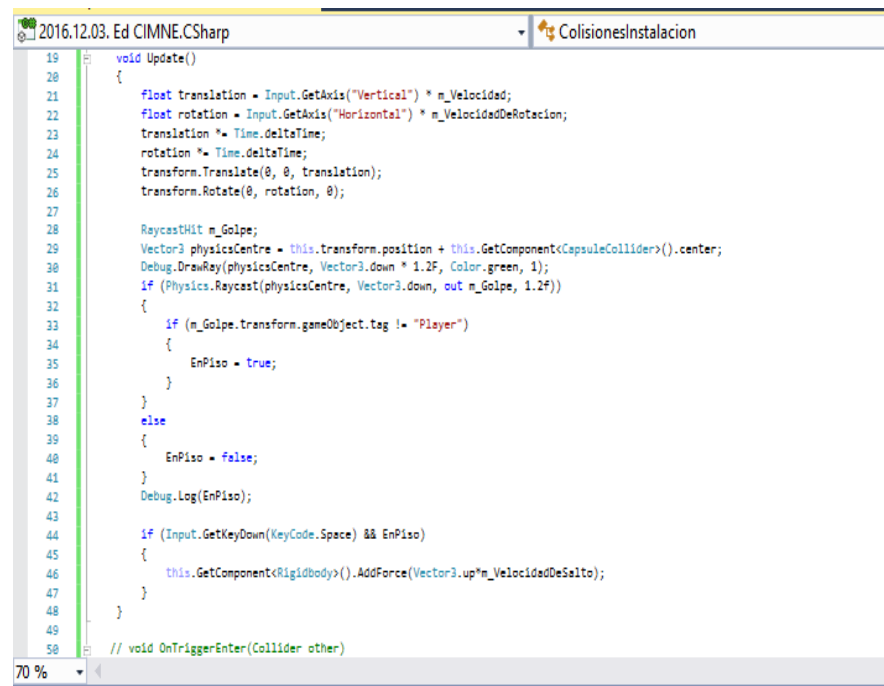


Figura 68. Ejemplo de código de programación en C Sharp para Rigidbody en Visual Studio para Unity 5

1.3. Is trigger: Con la creación de este script podremos ejecutar la orden de cuando queremos que un objeto entre en contacto con otro, es decir, este script se le asigna al objeto que va a recibir la colisión del personaje.

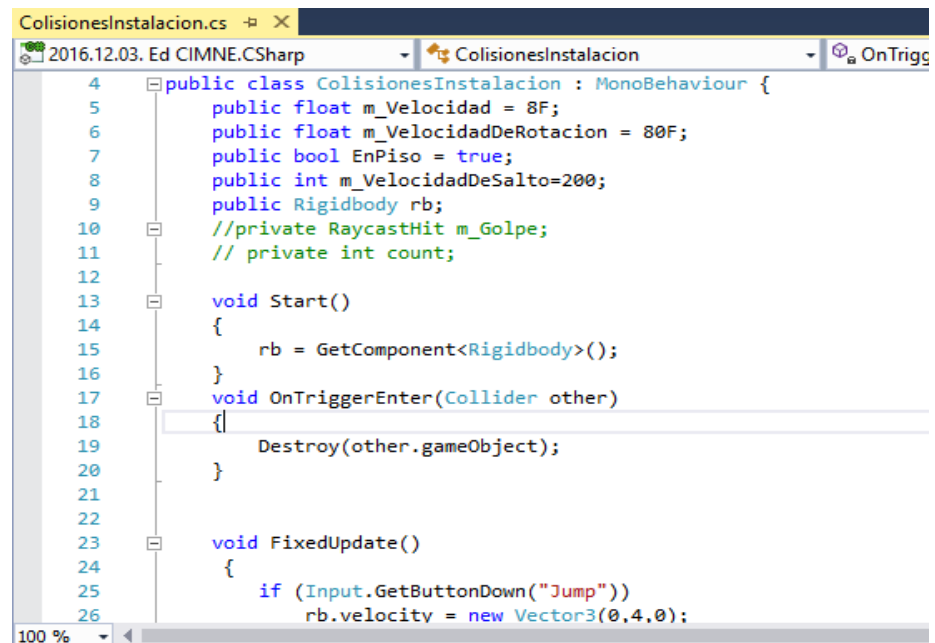


Figura 69. Ejemplo de código de programación en C Sharp para IsTrigger en Visual Studio para Unity 5

Para completar los scripts creados, es necesario que le FPS creado con anterioridad en la ventana de inspector tenga la etiqueta de jugador, ya que de lo contrario nuestra escena no funcionara.

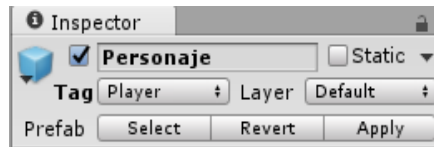


Figura 70. Propiedades de etiquetado en Unity 5

Además, en las propiedades de inspector también hay que activar la casilla de malla de colisión e IsTrigger de cada objeto donde se va colocar el script IsTrigger.

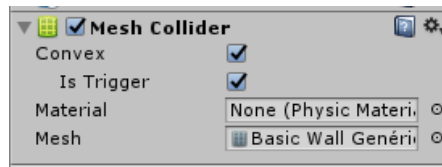


Figura 71. Propiedades de la malla en Unity 5

Como se puede ver en la figura 72 una vez ejecutado todos los script y conforme el personaje fue avanzando en la escena los elementos como por ejemplo el cristal central han sido ocultados para dejar solo a la vista el sistema de instalación.

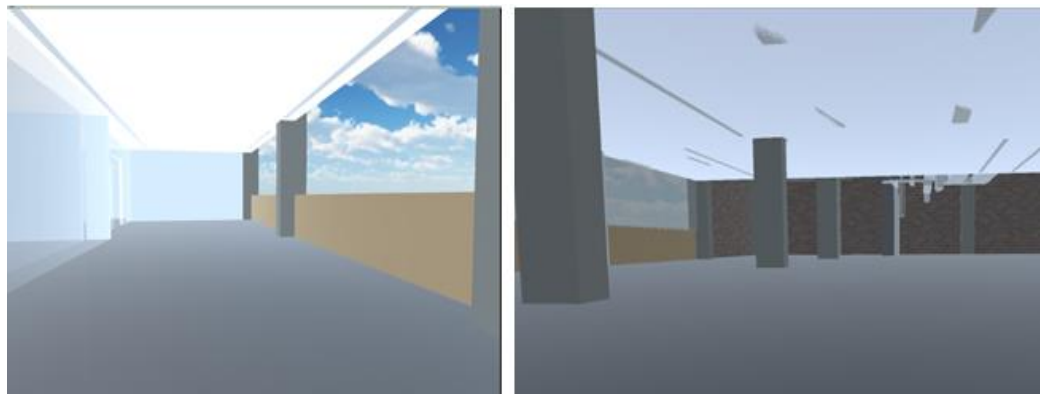


Figura 72. Comparación de imágenes al ejecutar los scripts en C Sharp para Unity 5

2. Creación de iluminación y ambiente: Es importante además tener en cuenta la trayectoria del sol y como este incide en el edificio ya que si nos fijamos con detalle la mayor parte del edificio se encuentra acristalado y por tanto habrá que tener en cuenta como el sol afecta a las instalaciones

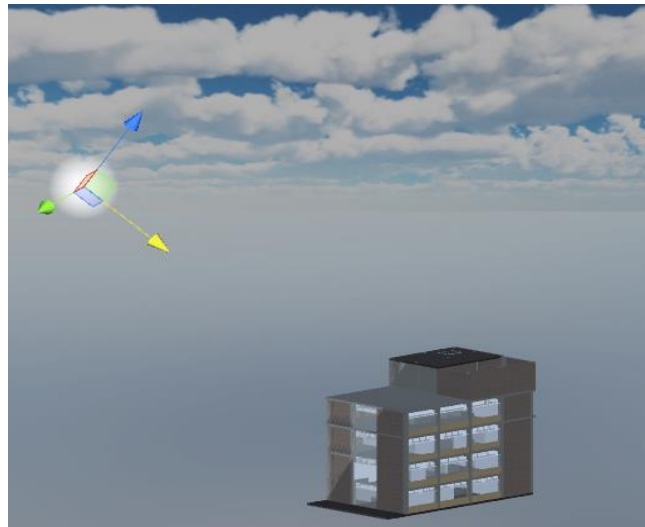


Figura 73. Efecto de la incidencia del sol sobre el edificio en Unity 5

Para crear la trayectoria del sol es necesaria la creación de una animación por tanto desde la ventana de inspección vamos a modificar las propiedades de la luz direccional donde a través de componentes vamos a elegir la opción animación, donde nos generara de manera automática una ventana nueva como aparece en la figura 74.

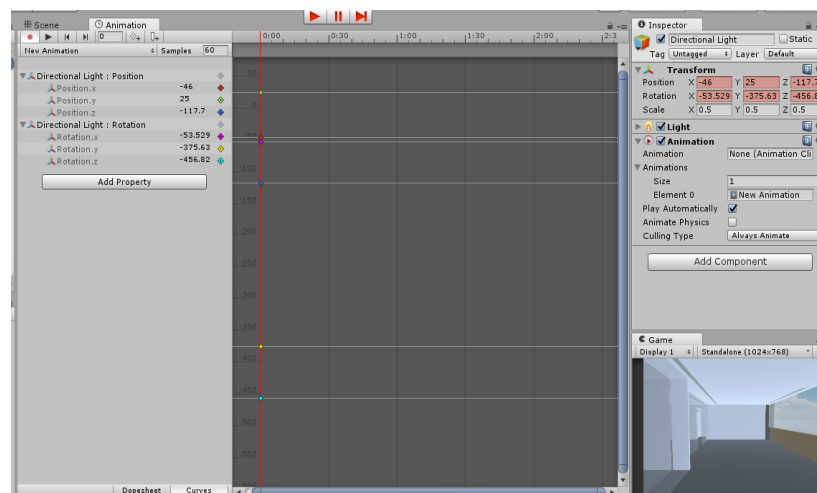


Figura 74. Ventana de propiedades de luz direccional con propiedades de animación para Unity 5

A partir de aquí vamos a trabajar en esta ventana donde si nos fijamos al rotar la trayectoria solar nos genera 3 Key Frames que irán referidos a los ejes de coordenadas globales

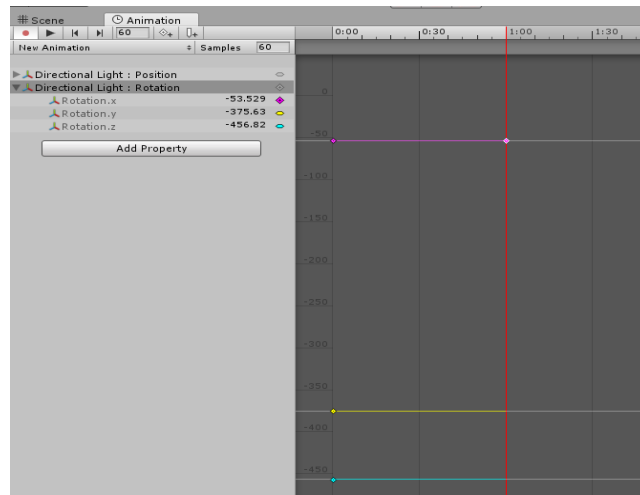


Figura 75. Propiedades de Key Frames por rotación en Unity 5

Cuando rotamos en los tres ejes si nos fijamos en la figura 76 las líneas de tiempo vemos que solo varía una y las otras dos son estáticas, esto quiere decir que la trayectoria del sol girara en torno a este eje que si nos fijamos en la figura 75 corresponde al eje x. Una vez conocido el eje de rotación, lo siguiente que se ha de hacer es definir el intervalo de tiempo de la misma, que nos dará una aproximación de la incidencia del sol sobre el edificio, nosotros lo hemos fijado en 60 segundos

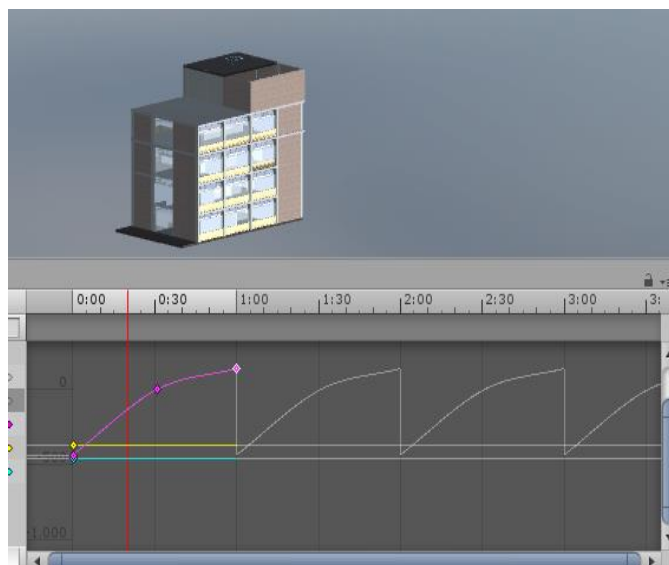


Figura 76. Trayectoria solar alrededor del eje x con una duración de 60 segundos para Unity 5

Definida la trayectoria solar a través de la ventana de propiedades del proyecto vamos a guardar la línea de tiempo creada o Frame. Una vez guardada, volvemos al proyecto general y en la ventana de inspector y dentro de la pestaña animación vamos a tener una opción llamada Animación, lo que hacemos a continuación es arrastrar nuestra animación guardada como new animation y colocarla sobre esta pestaña, tal y como se muestra en la figura 77.

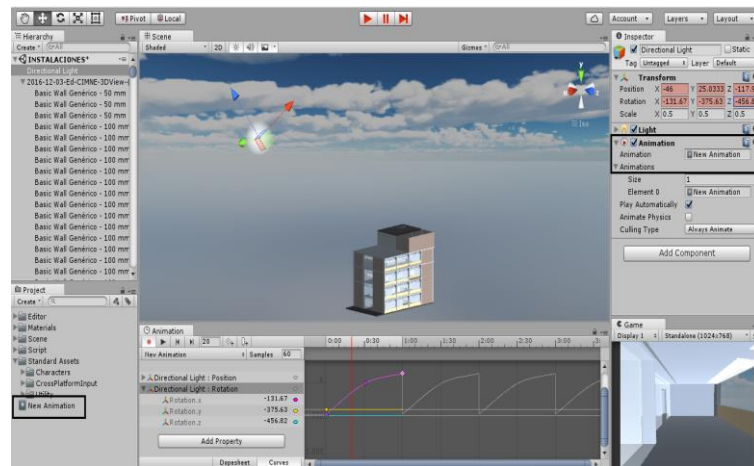


Figura 77. Propiedades de animación y definición de trayectoria para Unity 5

Para poder visualizar, la animación creada es muy importante seleccionar en la barra de propiedades el modo Loop. Este nos permitirá reproducir la animación creando así un bucle de tiempo que se iniciará con la puesta del sol y terminará con la ocultación de la misma.

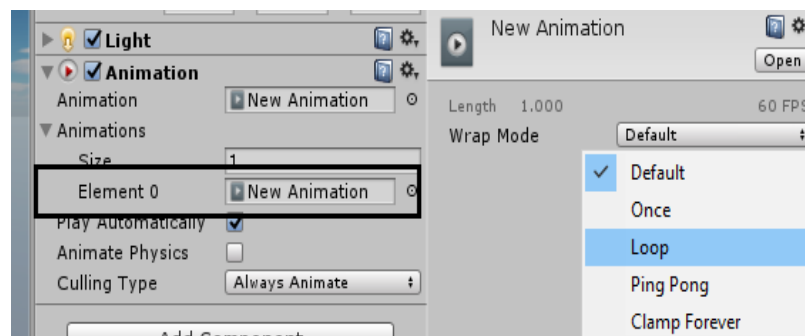


Figura 78. Colocación de la animación al elemento en Unity 5

Por último, se ha de probar y verificar que funciona de manera correcta. Como se puede ver en las figuras 79 y 80 todo es correcto.



Figura 79. Vista trayectoria del sol sobre el edificio para Unity 5



Figura 80. Vista trayectoria del sol sobre el edificio para Unity 5

- C. Resultado: Terminada la puesta a punto del edificio, podemos decir que todos los elementos de estudio marcados en la figura 14 se pueden comprobar. Donde la idea principal de búsqueda en este punto era dejar vista las instalaciones del edificio y ver la incidencia del sol en el mismo.

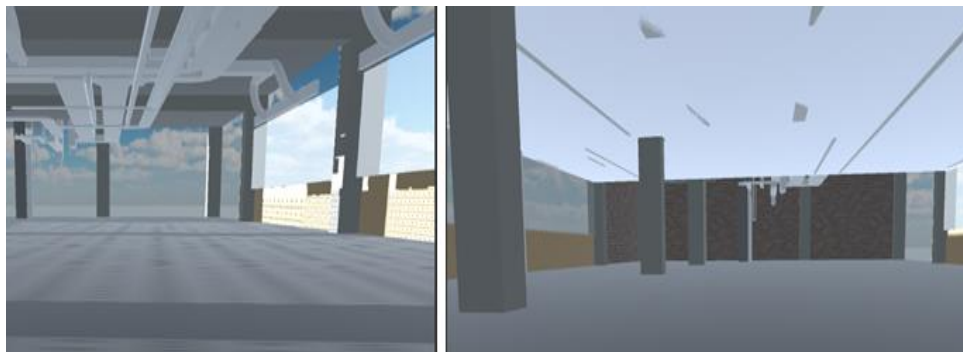


Figura 81. Visualización de la escena para Unity 5

4. Capítulo 4: Análisis del proyecto

4.1. Introducción

Una vez terminada la puesta a punto y la valoración de los dos casos de uso planteados en el capítulo 3, vamos a proceder a evaluar las distintas posibilidades, funcionalidades y sobre todo vamos a valorar de manera global si el trabajo que se ha llevado a cabo en este proyecto puede ser o no factible, valorando tanto los puntos positivos como negativos de su ejecución, dando además a conocer las distintas problemáticas planteadas a lo largo de la redacción del proyecto.

4.2. Desarrollo

Para su puesta a punto vamos a crear un nuevo proyecto en Unity 5 donde

1. En la carpeta de escenas vamos a guardar las dos escenas creadas para los casos de uso que se han planeado en el capítulo 4
2. Creamos una escena nueva llamada Menú. Esta escena será la base donde va a partir el modelo y los distintos submenús tal y como se puede ver en la figura 82.



Figura 82. Creación de la escena central (Menú) para Unity 5

3. Dado el grado de complejidad del proyecto, debido sobre todo a la gran cantidad de datos e información, es recomendable tener un orden previo de los elementos sé que van a ejecutar en Unity 5, por tanto, es bueno que en proyecto se creen carpetas donde podamos ordenar por ejemplo los materiales, script o assets que vienen por defecto en Unity 5

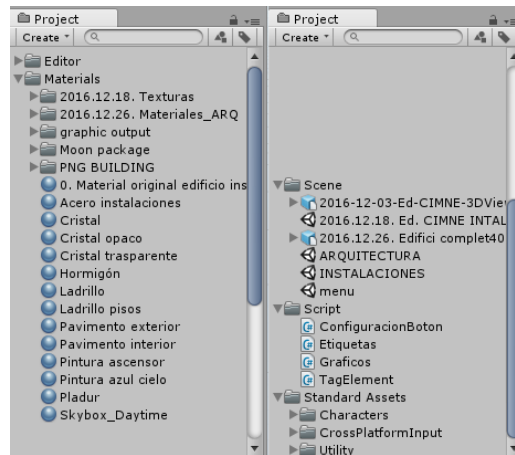


Figura 83. Orden de carpetas en Unity 5

4. Para movernos dentro de la escena es necesario crear un orden de los distintos escenarios de trabajo (Figura 84), para ellos iremos a la barra de propiedades o built setting y adjuntar las distintas escenas donde el orden será el siguiente
 - A. Escena de menú: Es la escena principal del proyecto, aquí visualizaremos los distintos submenús creados como por ejemplo la calidad gráfica o la visualización de planos de las distintas plantas del edificio. Este menú es importante ya que aquí irán también asociadas las escenas de los dos casos de uso planteado en el capítulo 4
 - B. Escena de Arquitectura: Ira asociado el caso de uso 1
 - C. Escena de Instalaciones: Ira asociado el caso de uso 2

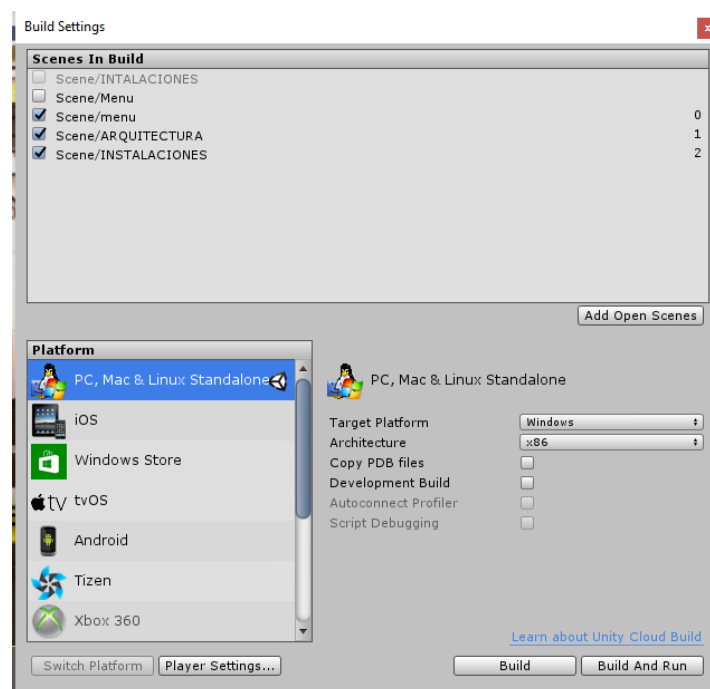


Figura 84. Jerarquización de las escenas en Unity 5

A partir de ahora todo el trabajo se va a realizar dentro de la escena menú, por tanto, vamos a ejecutar dicha escena y a colocarla en una vista en 2D, para facilitar el trabajo.

En la pestaña de inspección vamos a elegir una resolución de 1024x768, elegimos arbitrariamente esta ya que queremos que el menú se adapte a las distintas posiciones de la pantalla. Figura 86

A continuación, vamos a colocar un canvas, este nos ayudará a crear un recuadro o maco que corresponderá a la pantalla de la escena. Figura 85

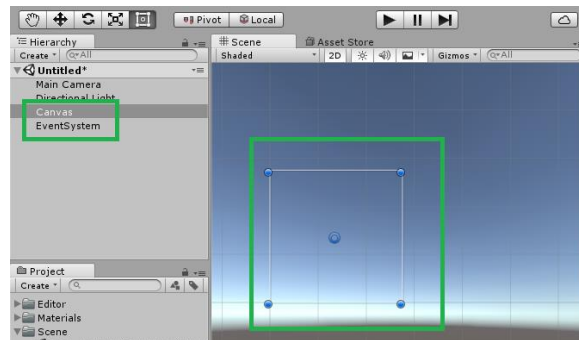


Figura 85. Creación de las propiedades del menú por canvas en Unity 5

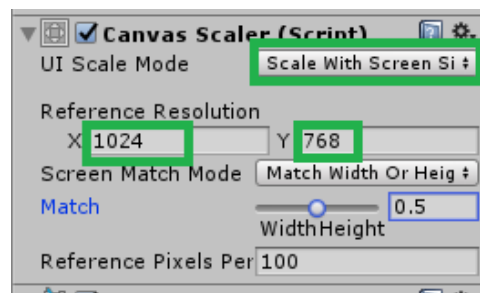


Figura 86. Propiedades características de menú en Unity 5

A partir de este Canvas ira referido todo el menú. Dentro de canvas vamos a crear un game object al que denominaremos menú, Es importante que dentro de Canvas se creen todos los demás submenús, para facilitar el trabajo se puede hacer a través del comando CTRL+D que se encarga de duplicar, a continuación, renombraremos cada uno de los creados donde la vista final será tal y como se observa en la figura 87

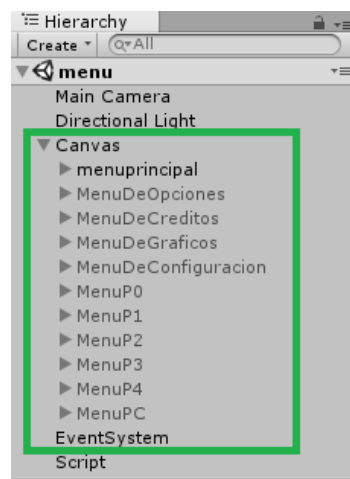


Figura 87. Creación de los submenús por medio de canvas en Unity 5

Al igual que en el capítulo 4 es necesario crear un script que me asocie todo lo anterior, es decir, tenemos que crear un comando que ejecute la orden de acceder a cada una de las funciones de menú. Figura 88

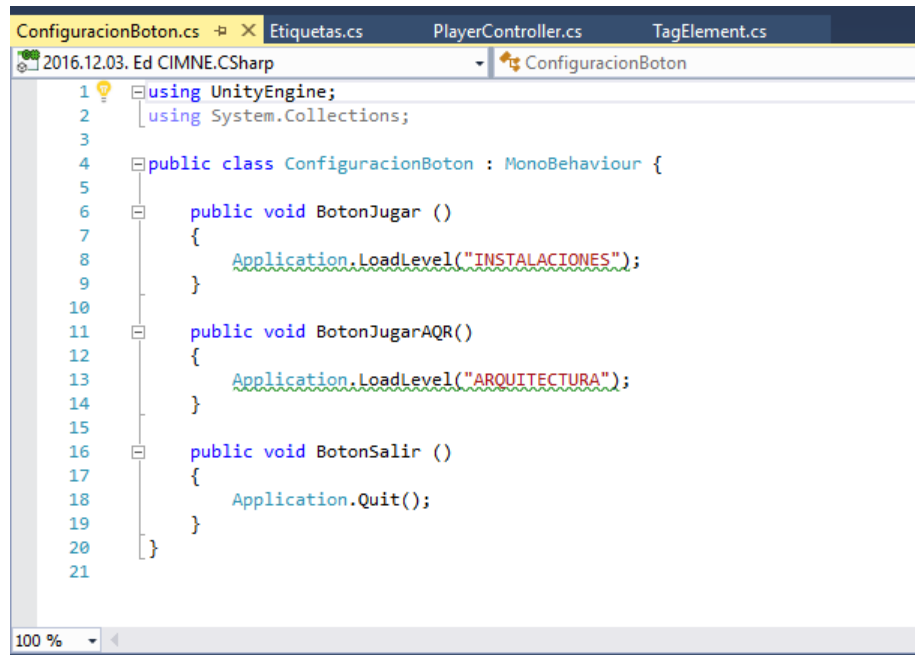


Figura 88. Ejemplo de código de programación en C Sharp para OpcionesDeMenu en Visual Studio para Unity 5

Personalmente diré que los scripts los creo conforme avance la ejecución del proyecto, ya que de esta manera se podrá entender de manera más claro los pasos que se han seguido.

Para crear cada botón de dentro de cada una de las opciones de submenú, lo único que se ha de hacer es seleccionar la carpeta donde se quiera crear, por ejemplo, seleccionamos menú principal, le damos un clic derecho – UI – BUTTOM, con el resto de carpetas se hará lo mismo. De esta forma se duplicarán y creará el menú de opciones y créditos o assistant. Figura 89.

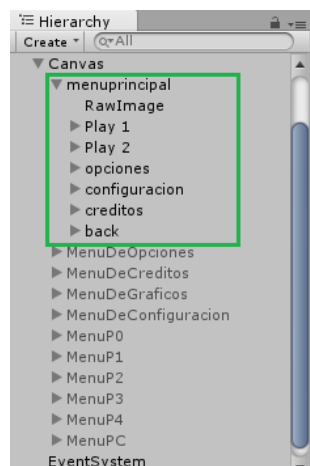


Figura 89. Propiedades características de los distintos botones dentro de cada menú para Unity 5

La opción de canvas nos permite además personalizar el menú solo basta ir a la pestaña de inspector y agregar una componente tipo UI – IMAGE con una textura tipo SPRIDE, esta es la encargada de dar color y forma al menú tal y como se puede ver en la figura 90.



Figura 90. Visualización final de menú para Unity 5

Otro de los scripts que crearemos será referido a los gráficos ya que dependiendo de la potencia del ordenador la visualización se puede ver ralentizada, por ello se ha creado conveniente crear 6 niveles de gráficos, cada uno de ellos va a ir asociado a un nivel de calidad gráfica que puede ir desde calidad muy baja a excelente. Figura 91

```

Graficos.cs  ConfiguracionBoton.cs*
2016.12.03. Ed CIMNE.CSharp  Graficos  Graficos2()

1  using UnityEngine;
2  using System.Collections;
3
4  public class Graficos : MonoBehaviour
5  {
6      public void Graficos1()
7      {
8          QualitySettings.currentLevel = QualityLevel.Fastest;
9      }
10
11     public void Graficos2()
12     {
13         QualitySettings.currentLevel = QualityLevel.Fast;
14     }
15
16     public void Graficos3()
17     {
18         QualitySettings.currentLevel = QualityLevel.Simple;
19     }
20
21     public void Graficos4()
22     {
23         QualitySettings.currentLevel = QualityLevel.Good;
24     }
25
26     public void Graficos5()
27     {
28         QualitySettings.currentLevel = QualityLevel.Beautiful;
29     }
30
31     public void Graficos6()
32     {
33         QualitySettings.currentLevel = QualityLevel.Fantastic;
34     }
35

```

Figura 91. Ejemplo de código de programación en C Sharp para CódigoDe Visualización en Visual Studio para Unity 5

Terminada su creación simplemente arrastrando con el ratón llevamos este menú a la pestaña de jerarquía. Asignamos cada script con su gráfico es decir seleccionamos play 1 y dentro de la pestaña de inspector vamos a “on click” agregamos un nuevo componente seleccionamos “no función gráficos” y finalmente seleccionamos gráfico 1(). Figura 92



Figura 92. Ejecución de las opciones de gráficos para Unity 5

Para comprobar que todo funciona de manera idónea, vamos a probar, para ello clicamos en botón de play en la barra principal de la barra de propiedades de Unity 5. Como se puede ver en la figura 93 hemos asociado una calidad gráfica baja mientras que en la figura 94 está asociada a una calidad gráfica alta, esto como se puede observar tendrá un efecto negativo en mi ordenador y es que el nivel de recorrido se vio afectado debido a la calidad y capacidad grafica de mi ordenador.

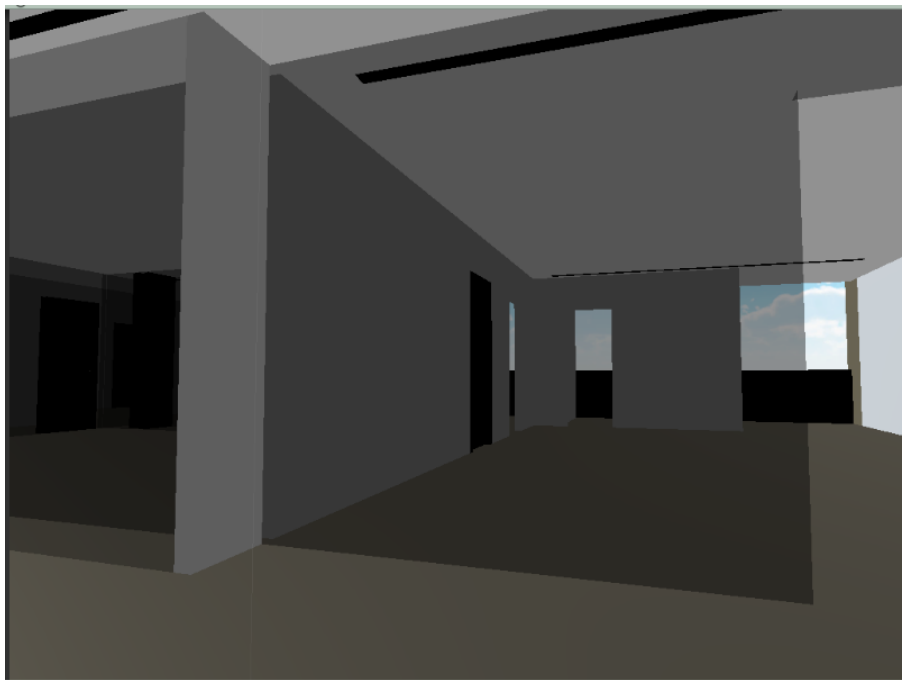


Figura 93. Calidad baja en Instalaciones para Unity 5



Figura 94. Calidad óptima en Arquitectura para Unity 5

Siguiendo con el orden de creación de nuestro menú, otra de las pestañas que vamos a encontrar dentro es la referente a los planos, esta es muy importante ya que nos servirá para comprobar, por ejemplo, si lo realizado en la obra corresponde a los acordado al inicio del proyecto o para visualizar que tipo de elementos y en qué lugar de la obra están colocados

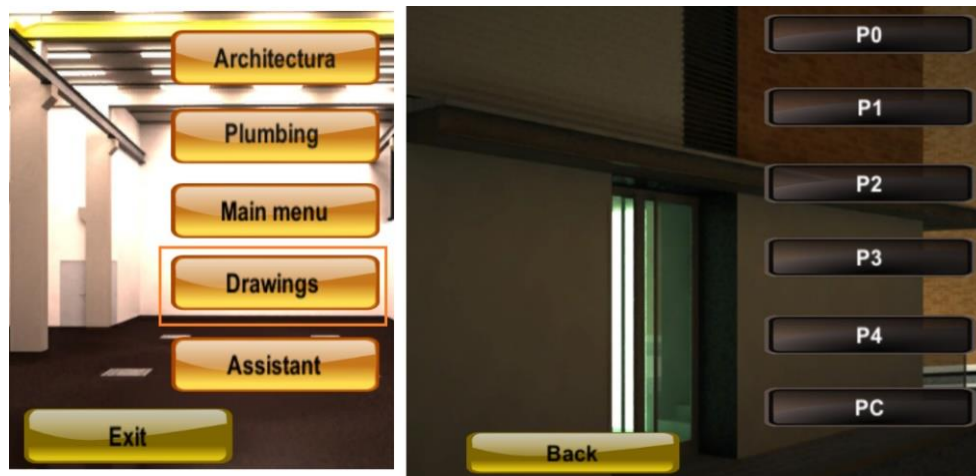


Figura 95. Menú y propiedades de los planos para Unity 5



Figura 96. Visualización del plano asociado a la primera planta del edificio B0. CIMNE. UPC Campus Nord para Unity 5



Figura 97. Visualización en 3D de la primera planta del ed. B0. CIMNE. UPC Campus Nord para Unity 5

Como se puede comprobar el plano coincide con la visualización, por tanto, en una primera comprobación podemos decir, por ejemplo, que se cumple con los requisitos marcados en los planos iniciales. Pero si pasamos a comprobar la misma planta en la escena de instalación no coincide. Por tanto, esta información debería de constar en el plano actual con la colocación de una nota de llamado o modificar y actualizar el plano.



Figura 98. Visualización de la primera planta del ed. B0. CIMNE. UPC Campus Nord para Unity 5

4.3. Valoración y resultados

Una vez se ha finalizado el proyecto vamos a proceder a valorar los resultados obtenidos del mismo. Si hacemos una primera valoración podemos ver que una de las desventajas que se tiene al trabajar con la unión Revit – Unity 5, es la pérdida de materiales y de texturas, esto es debido a que el archivo fue modelado, diseñado y proyectado con los propios materiales que viene por defecto en Revit.

Si comparamos ambos casos de uso se puede ver que la ventaja que posee el caso de uso 1 con respecto al caso de uso 2 es la carpeta de materiales, el primero estará basado en los materiales reales del edificio, mientras que el segundo será solo una aproximación ya que todas las texturas fueron creadas directamente desde Revit.

En Unity 5 la colocación de materiales también lleva un tiempo y proceso de trabajo ya que no es lo mismo crear un material como si fuera un sólido de un color como por ejemplo el color gris de una puerta o un material con características especiales como por ejemplo los ladrillo o los muros de grava, donde se ha de buscar la imagen en formato JPG se crea una subcarpeta dentro de la carpeta de materiales y se siguen los mismos pasos como si fuera un material sólido, es decir, es la ventana de inspector si clicamos dos veces en el símbolo de Albedo, nos dejara asignar el JPG como un archivo en formato meta al material que se ha creado.

Pero hay que decir que debido a la incompatibilidad de versiones del programa muchas de sus propiedades características se perdieron, como por ejemplo las texturas. En mi caso no supuso un problema ya que lo que realmente me interesaba de este archivo además del modelo fue su carpeta de materiales, pero si el archivo hubiera tenido que modelarse de 3ds Max sí que me hubiera supuesto un trabajo extra ya que al contrario de Unity 5 en 3ds Max es necesario primero ver el tipo y propiedades de mapeado de texturas, luego asignarle el material y para finalizar crear la animación por medio de V-ray o mental ray.

De los resultados obtenidos hasta el momento se puede decir que, primero trabajar con el primer método que se ha planteado en los esquemas de las figuras 14, 15, 16 y 18, presentan una serie de ventajas en cuanto a la versatilidad que presenta a la hora de diseñar el edificio, ya que como hemos partido de la idea de crear un edificio a imagen y semejanza del real pero simplificando el trabajo se hace más fácil ya que trabajar primero con AutoCAD nos estamos asegurando la importación de aquellos elementos que son importantes a la hora de modelar este en Revit, como las unidades y los elementos tanto estructurales como arquitectónicos que posee el edificio. Otra de las ventajas que posee es que toda la información que se crea en el modelo ya se actualiza de manera automática, gracias a la interoperabilidad que posee la conexión del programa con BIM, además podemos crear una serie de plantillas que nos van a aportar información extra y como complemento podemos hacer un recorrido en primera persona y ver cómo quedaría la obra hasta el momento. Por otra parte, hay una conexión directa entre Revit (FBX) y Unity (FBX), ya que ambas trabajan en el mismo formato, esto nos supondrá una ventaja, ya que nos saltamos el paso de vincular el archivo primero a 3ds Max.

En cuanto a la evaluación de las instalaciones una de las conclusiones que obtenemos es que al dejar solo la instalación vista podemos ser capaces de detectar algún tipo de error que se pueda producir antes de llevar el modelo a obra o por ejemplo un caso más real comprobar como incide la trayectoria del sol en el edificio, ya que en la actualidad este es uno de los problemas a los que se enfrenta en departamento de CIMNE. Una de las conclusiones que se ha planteado es la colocación de lamas o cortinas que impidan el paso de la luz solar.

5. Capítulo 5: Conclusiones

Estudiar nuevas metodologías de trabajo que sean compatibles con las formas de trabajo tradicional dentro de cada sector es muy complejo, pero no difícil de compatibilizar, como se ha visto a lo largo de la redacción y explicación de este documento, donde de manera práctica y teórica se ha explicado la conexión que tiene el mundo relacionado con entornos multiplataforma con otro totalmente distinto que es el mundo de la construcción asociado a un modelo BIM. Para poder implementar la unión fue necesario tener un software compatible, como Unity 5 con el hemos visto que podemos recrear esa realidad virtual que tanto buscamos a través de la recreación de un escenario mediante herramientas informáticas, llevándonos así por tanto a esa inmersión y conexión ansiada

A nivel de implementación hemos partido de dos modelos ya existentes, uno diseñado en formato Revit que nos sirvió como un modelo As Built donde analizamos sus instalaciones, fontanería y también para la comparación de modelos y otro en formato 3ds Max, donde se representó el diseño conceptual del módulo que nos sirvió para el estudio arquitectónico y estructural. Cada una de estas dos representaciones del edificio nos sirvió además para presentar los dos casos de uso planteados donde en el primero exploró la posibilidad que ofrece el recorrido virtual para la construcción, revisión, etiquetado virtual de espacio y para la comparación de planos efectos y de iluminación.

El segundo modelo se sirvió para la revisión de las instalaciones como medio de información y alerta ante cualquier problema de solapamiento en obra. La conexión entre ambos se maneja en torno a un formato que es el FBX. Este no es un formato propio de los modelos BIM, pero eso no quiere decir que no sea compatible ya que al evaluar el modelo que se tenía en Revit en el menú de propiedades del mismo Revit, nos da como una opción de guardado.

Si analizamos de manera global los resultados obtenidos se puede decir que crear un modelo virtual nos va a poder dar una idea anticipada de cómo se verá la estructura una vez finalice su proceso de construcción.

Otra de las ventajas del uso de esta tecnología es la capacidad para poder visualizar con anticipación, poder evaluar durante la etapa constructiva hasta qué punto se está cumpliendo lo marcado en las condiciones iniciales del proyecto, avisarnos de algún tipo de problema o solapamiento de trabajo en obra o, por ejemplo, vamos a ser capaces de predecir cómo va a incidir el sol en las fachadas exteriores del edificio. Este último aspecto, de hecho, ha sido un punto de inflexión y debate, ya que cuando se proyectó la construcción del edificio no se tuvo en cuenta la incidencia solar, que actualmente resulta molesta a los ocupantes. Otro ejemplo práctico del interés práctico de uso de estas herramientas es advertir las diferencias en cuanto a sus ejecuciones iniciales respecto a la construcción final. En particular, en la etapa de proyección figura una escalera central que une la planta baja con la primera planta, de acuerdo con el modelo de 3ds Max, mientras que en el modelo de Revit, la información As-Built del edificio, tal escalera ya no existe. Es esencial que este cambio figure en la memoria adjunta al proyecto, o estar señalada mediante algún tipo de alerta en los planos, convirtiéndose así en un punto crítico delicado desde el punto de vista de gestión y documentación de la obra.

Llegado a este punto y después de ver las ventajas que se tiene al combinar las formas de trabajo de ambos modelos puedo afirmar que su unión es factible y posible ya que, por ejemplo, si esta compatibilidad de formatos se hubiera ejecutado previa a la construcción del edificio nos habríamos podido dar cuenta de la importancia que tiene la construcción de lamas para evitar el paso de luz solar en sus horas críticas.

Como conclusión final puedo decir además que el coste de implantación de este proyecto no es tan alto como muchos pueden llegar a pensar, si hacemos unos cálculos rápidos solo basta con multiplicar el número de hora de trabajo de una persona que tenga formación tanto de Unity 5 como de cualquier otro programa de modelado BIM * el número de horas diaria de trabajo * el número de meses/semanas /días de ejecución del mismo y tendremos un coste, que además es mucho más barato que un error de ejecución o proyección en obra que nos puede llevar a disparar los gastos y elevar el presupuesto inicial acordado.

Esto se puede ver claramente en los remates finales que se hacen en las obras una vez están terminadas, solo en arreglar los remates supone alrededor de un 20% de total, esto es mucho dinero que no podemos ni debemos gastar ya que, si no, el beneficio que obtendríamos sería mucho menor e incluso en algunos casos podríamos salir perdiendo. Por tanto, implantar la combinación de realidad virtual - entornos BIM es necesaria.

6. Capítulo 6: Futuras líneas de trabajo

Aunque se han dado grandes pasos en el estudio y conexión de BIM con la realidad virtual, aún queda mucho camino por recorrer ya que lo que hemos hecho en este proyecto es apenas una pincelada de lo que se puede llegar hacer y descubrir con estas herramientas que además poseen una magnífica conexión.

Una de las líneas de trabajo que queda abiertas es poder llegar a conseguir que nuestro modelo se pueda conectar con un Smartphone que nos permita ver de manera más sencilla y cómoda la evolución de la obra, que nos avise de posibles errores o simplemente nos aporte información extra.

Otra de las grandes líneas de estudio que puede llegar a tener gran interés no solo a nivel académico si no también en el trabajo diario, es poder conectar con la realidad aumentada. Una de las posibles grandes ventajas que nos traería esta conexión sería poder visualizar toda la información que se tiene hasta el momento y actualizarla a tiempo real obteniéndose así la actualización de todos los datos donde constaran las posibles modificaciones.

Si se llegase a conseguir sería una herramienta con un gran potencial, ya que nos sería de gran utilidad a la hora de realizar nuestro trabajo. Uno de los beneficios donde podría verse reflejado es en la eficiencia y eficacia del trabajo con el consecuente ahorro que se vería reflejado en el margen de beneficio que cada proyecto posee.

7. Referencias

- [1] UNITY INC, «Unity,» [En línea]. Available: <https://unity3d.com/es/unity>. [Último acceso: 13 Junio 2016].
- [2] Unity (software), «Wikipedia,» [En línea]. Available: [https://es.wikipedia.org/wiki/Unity_\(software\)](https://es.wikipedia.org/wiki/Unity_(software)). [Último acceso: 13 Junio 2016].
- [3] App Store Unity, «UNITY INC,» [En línea]. Available: <https://www.assetstore.unity3d.com/en/>. [Último acceso: 13 Junio 2016].
- [4] Hildebrandt Gruppe, «Hildebrandt Gruppe,» [En línea]. Available: <http://www.hildebrandt.cl/que-es-revit-y-para-que-sirve-en-el-modelado-bim/>. [Último acceso: 15 Junio 2016].
- [5] 24 Studio - Realidad Virtual – BIM , el gran cambio inmobiliario, «24studio in BIM, Consultoría, Implementación, VIZ, VR,» 13 Mayo 2016. [En línea]. Available: <http://www.24studiolab.com/realidad-virtual-bim/>. [Último acceso: 21 Junio 2016].
- [6] Building Information Modeling, «Wikipedia,» 20 Junio 2016. [En línea]. Available: https://es.wikipedia.org/wiki/Modelado_de_informaci%C3%B3n_de_construcci%C3%B3n.
- [7] Las dimensiones del BIM, «Antonio Flores Arquitectura,» [En línea]. Available: <https://antoniofloresarquitectura.wordpress.com/2015/07/15/las-dimensiones-del-bim/>. [Último acceso: 21 Junio 2016].
- [8] Graphisoft Open BIM, «Graphisoft,» [En línea]. Available: https://www.graphisoft.es/archicad/open_bim/about_bim/. [Último acceso: 20 Junio 2016].
- [9] Breve introducción a los niveles LOD en modelos BIM , «Sánchez Matamoros Arquitectos,» 14 January 2015. [En línea]. Available: <http://www.sanchez-matamoros.com/blog/2015/1/14/breve-introduccion-a-los-niveles-lod-en-modelos-bim>. [Último acceso: 21 Junio 2016].
- [10] B. Community, Interviewee, *El caso de 'BIM para Terminales de Graneles Sólidos', de ARUP España*. [Entrevista]. 28 Abril 2016.
- [11] P. J. Abós, «Realidad aumentada en arquitectura mediante modelado 3D,» 23 Mayo 2014. [En línea]. Available: <http://pilarjimenezabos.com/realidad-aumentada-en-arquitectura/>. [Último acceso: 21 Junio 2016].
- [12] UPC Commons, «Location based augmented reality application on Unity 3D,» [En línea]. Available: <http://hdl.handle.net/2099.1/19709> . [Último acceso: 14 Junio 2016].
- [13] Wikipedia, «Autodesk Revit Wikipedia,» [En línea]. Available: <https://es.wikipedia.org/wiki/Revit>. [Último acceso: 15 12 2016].
- [14] Wikipedia, «Autodesk 3ds max studio,» [En línea]. Available: https://es.wikipedia.org/wiki/Autodesk_3ds_Max. [Último acceso: 15 12 2016].
- [15] Realidad Virtual y entornos 3D, «Universidad de Coruña,» [En línea]. Available: <http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/3D/Realidad%20Virtual/web/definicion.html>. [Último acceso: 16 12 2016].
- [16] Realidad virtual, «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Realidad_virtual. [Último acceso: 16 12 2016].
- [17] Qué es la realidad virtual, «Realidad virtual.com,» [En línea]. Available: <http://www.realidadvirtual.com/que-es-la-realidad-virtual.htm>. [Último acceso: 16 12 2016].

- [18] Realidad Aumentada (AR), «Monografias.com,» [En línea]. Available: <http://www.monografias.com/trabajos-pdf5/monografia-realidad-aumentada/monografia-realidad-aumentada.shtml>. [Último acceso: 17 12 2016].
- [19] Bimetriclab, «Efecto del BIM en las fases del ciclo de vida de un proyecto,» Espacio Lean BIM. Construcción colaborativa, 27 Abril 2016. [En línea]. Available: <http://www.espacioleanbim.com/episodio-10-efecto-del-bim-las-fases-del-ciclo-vida-proyecto/>. [Último acceso: 21 Junio 2016].
- [20] 3DS Max, «AutoDesk,» [En línea]. Available: <http://www.autodesk.com/products/3ds-max/overview>. [Último acceso: 20 01 2017].
- [21] Taringa, «¿Que es el 3D?,» 2010. [En línea]. Available: <https://www.taringa.net/post/info/7317756/Que-es-el-3D-definiciones.html>. [Último acceso: 20 01 2017].
- [22] 4M BIM, «BIM España,» [En línea]. Available: <http://www.es.4mbim.com/what-is-bim/>. [Último acceso: 20 Junio 2016].

Índice de figuras

Figura 1. Esquema colaborativo: Caso de uso1 . Arquitectura + Estructura.....	13
Figura 2. Esquema colaborativo: Caso de uso 2. Instalación	13
Figura 3. Comparación de los distintos formatos de Trabajo.....	14
Figura 4. Funcionamiento de un flujo BIM [AMBIM Arquitectura e ingeniería]	16
Figura 5. Fases del ciclo de vida del Proyecto en Etapa BIM 1 – proceso lineal.....	16
Figura 6. Fases del ciclo de vida del Proyecto en Etapa BIM 2 – modelo lineal	17
Figura 7. Fases del ciclo de vida del Proyecto en Etapa BIM 3 – modelo lineal	17
Figura 8. Ventajas en la comparación de trabajo Revit vs AutoCAD [CEO de HOK 2008].....	18
Figura 9. Clasificación del dimensionamiento [BIM Barcelona].....	19
Figura 10. Optimización de los recursos BIM vs CAD [Arquirapados]	19
Figura 11. Niveles de Level Of Developed en España [ARUP España] [3]	20
Figura 12. Desarrollo del nivel de madurez de un IFC [Keenllside, Llebich&Grobber 2012].....	21
Figura 13. Ficha técnica. Caso 1.....	28
Figura 14. Ficha técnica. Caso 2.....	28
Figura 15. Ficha técnica: Características específicas de 3ds Max.....	29
Figura 16. Ficha técnica: Características específicas de Unity 5	31
Figura 17. Estructura organizativa en unity 5.....	32
Figura 18. Estructura organizativa en Revit	33
Figura 19. Orden cronológico de trabajo. Caso 1	33
Figura 20. Perspectiva del modelo final en 3ds Max	34
Figura 21. Propiedades de formato de exportación en 3ds Max	34
Figura 22. Propiedades de los materiales en Unity 5	35
Figura 23. Asignación de material al objeto ladrillo en fachada en Unity 5	35
Figura 24. Propiedades características de luz.....	35
Figura 25. Propiedades características de las componentes	36
Figura 26. Vista interior de la planta baja del Ed. B0. CIMNE. UPC Campus Nord	36
Figura 27. Visualización de la planta sótano	36
Figura 28. Visualización exterior del Campus Nord UPC en Unity 5	37
Figura 29. Visualización exterior del Ed. B0. CIMNE. Campus Nord con Unity 3d	37
Figura 30. Ejemplo de código de programación en C Sharp para puerta simple en Visual Studio.....	38
Figura 31. Ejemplo de código de programación en C Sharp para puerta doble en Visual Studio.....	38
Figura 32. Bisagra como padre en Unity 5	39
Figura 33. Puerta como padre en Unity 5	39
Figura 34. Propiedades de desactivación de mallado en Unity 5	39
Figura 35. Creación de las colisiones en Unity 5	40
Figura 36. Orden jerarquía de trabajo en Unity 5.....	40
Figura 37. Propiedades específicas de mallado en Unity 5	40
Figura 38. Propiedades específicas de colisión en Unity 5	41
Figura 39. Propiedades específicas de script en Unity 5	41
Figura 40. Función tarjes. Caso cuando bisagra actúa como padre en Unity 5.....	41
Figura 41. Función tarjes. Caso cuando la puerta actúa como padre en Unity 5	41
Figura 42. Visualización en Unity 5	42
Figura 43. Visualización puerta lateral P1 en Unity 5.....	42

Figura 44. Visualización puerta trasera P1 en Unity 5	42
Figura 45. Ejemplo de código de programación en C Sharp para ClickDeMouse en Visual Studio	43
Figura 46. Ejemplo de código de programación en C Sharp para ScreenReceiver En Visual Studio	43
Figura 47. Creación de la colisión en Unity 5	44
Figura 48. Propiedades de la caja de colisión en Unity 5	44
Figura 49. Ejemplo de código de programación en C Sharp para etiquetado en Visual Studio.....	44
Figura 50. Colocación del elemento colisión en pilar extremo en Unity 5	45
Figura 51. Propiedades de mensaje en Unity 5	45
Figura 52. Resultado final de etiqueta en Unity 5	45
Figura 53. Mensaje. Etiquetado pilares en Unity 5	46
Figura 54. Mensaje. Etiquetado puertas en Unity 5	46
Figura 55. Mensaje. Etiquetado ventanas en Unity 5	46
Figura 56. Vista exterior. Ed.B0. CIMNE. Campus Nord para Unity 5.....	47
Figura 57. Orden cronológico de trabajo. Caso.....	48
Figura 58. Edificio modelado en Revit.....	48
Figura 59. Comprobación: Escala de trabajo en Revit	48
Figura 60. Comprobación: Coordenadas del proyecto en Revit.....	48
Figura 61. Proceso de formato importación Revit – Unity 5	49
Figura 62. Ruta del proyecto creada en Unity 5	49
Figura 63. Vista inicial del edificio en Unity 5	49
Figura 64. Vista final del edificio en Unity 5	50
Figura 65. Posición del FPS en la 2 planta del	50
Figura 66. Ejemplo de código de programación en C Sharp para colisión en Visual Studio para Unity 5	51
Figura 67. Rayo de detección de la escena en Unity 5	51
Figura 68. Ejemplo de código de programación en C Sharp para Rigidbody en Visual Studio para Unity 5	52
Figura 69. Ejemplo de código de programación en C Sharp para OnTrigger en Visual Studio para Unity 5	52
Figura 70. Propiedades de etiquetado.....	53
Figura 71. Propiedades de la malla en Unity 5	53
Figura 72. Comparación de imágenes al ejecutar los scripts en.....	53
Figura 73. Efecto de la incidencia del sol sobre el edificio en Unity 5	54
Figura 74. Ventana de propiedades de luz direccional con propiedades de animación para Unity 5	54
Figura 75. Propiedades de Key Frames por rotación en Unity 5.....	55
Figura 76. Trayectoria solar alrededor del eje x con una duración de 60 segundos para Unity 5	55
Figura 77. Propiedades de animación y definición de trayectoria para Unity 5.....	56
Figura 78. Colocación de la animación al elemento en Unity 5	56
Figura 79. Vista trayectoria del sol sobre el edificio para Unity 5	56
Figura 80. Vista trayectoria del sol sobre el edificio para Unity 5	57
Figura 81. Visualización de la escena para Unity 5	57
Figura 82. Creación de la escena central (Menú) para Unity 5	58
Figura 83. Orden de carpetas en Unity 5	59
Figura 84. Jerarquización de las escenas en Unity 5	59
Figura 85. Creación de las propiedades del menú por canvas en Unity 5	60

Figura 86. Propiedades características de menú en Unity 5	60
Figura 87. Creación de los submenús por medio de canvas en Unity 5	60
Figura 88. Ejemplo de código de programación en C Sharp para OpcionesDeMenu en Visual Studio para Unity 5	61
Figura 89. Propiedades características de los distintos botones dentro de cada menú para Unity 5	61
Figura 90. Visualización final de menú para Unity 5	62
Figura 91. Ejemplo de código de programación en C Sharp para CodigoDeVisualizacion en Visual Studio para Unity 5	62
Figura 92. Ejecución de las opciones de gráficos para Unity 5	63
Figura 93. Calidad baja en Instalaciones para Unity 5	63
Figura 94. Calidad óptima en Arquitectura para Unity 5	64
Figura 95. Menú y propiedades de los planos para Unity 5	64
Figura 96. Visualización del plano asociado a la primera planta del edificio B0. CIMNE. UPC Campus Nord para Unity 5	65
Figura 97. Visualización en 3D de la primera planta del ed. B0. CIMNE. UPC Campus Nord para Unity 5	65
Figura 98. Visualización de la primera planta del ed. B0. CIMNE. UPC Campus Nord para Unity 5	65

Glosario

Realidad aumentada: Se centra en la visualización de un entorno físico del mundo real que se asocia a elementos virtuales, pero a tiempo y en lugares reales
Realidad virtual: Es la construcción una realidad que no existe. Es un camino que tienen los humanos para visualizar manipular e interactuar por medio de un ordenador
BIM: Se define como la creación de procesos de generación y gestión de datos del proyecto durante su ciclo de vida utilizando software dinámico en 3D y a tiempo real.
As Built: También conocido como Proyecto conforme a obra. Se define como el proyecto de ingeniería referido a los planos, cálculos y descripciones de las actualizaciones que reflejan la adaptación del Proyecto de Ejecución a la realidad de la obra, los cambios pedidos durante el transcurso de la misma y de la construcción final del edificio, local o nave en cuestión.
FBX: Es un dibujo 2D o 3D guardados en el formato de Autodesk FBX. Se mantiene la fidelidad total y la funcionalidad del archivo original y puede ser manipulado por múltiples programas. Se utiliza para la creación de la interoperabilidad entre aplicaciones 3D.
MEP: Revit está compuesto por 3 disciplinas que son Arquitectura, Estructura y MEP. Esta última, además se la conoce como Sistema o Instalaciones, ya que las siglas M.E.P. viene de las palabras en inglés Mechanical, Electrical y Plumbing. Es decir, Mecánica, Eléctrica y Fontanería
CAD: Proviene del nombre de Computer-Aided Design. Se define como el uso de programas informáticos para crear representaciones gráficas de los objetos físicos en dos o tres dimensiones. El software CAD puede ser especializado para aplicaciones específicas.
ASSET: Cuando usted cree un proyecto de Unity, usted está creando una carpeta llamada My Unity Project, que contiene las siguientes subcarpetas: Assets, Library, Obj y Project Setting. La carpeta de Assets es donde usted debería guardar o copiar los archivos que usted quiera utilizar en su proyecto.
OCULUS RIFT: Las Oculus, son cascos de realidad virtual que recrean un visón en 3D a gran profundidad creando una sensación de inmersión total, por ejemplo, en un video juego o en mi caso en la inspección de obra. También se puede conectar a los móviles y al PC
INDUSTRY FOUNDATION CLASSES (IFC): Los formatos IFC, son los más utilizados para entornos de trabajo multitarea, son como, por ejemplo .DOC para Word, para revit uno de las formas en que se puede guardar es en IFC y por ejemplo con este formato podemos abrirlo en unity y no hay ningún problema de incompatibilidades.
SCRIPT: Se define como un documento que contiene instrucciones, escritas en códigos de programación, es decir, el script es un lenguaje de programación que ejecuta diversas funciones en el interior de un programa informático.
FPS: Se define como un objeto que permitirá manejar un personaje en la pantalla, es decir, posibilita la visualización del entorno desde la visión del personaje
V-ray: Se define como es un motor de renderizado usado como extensión para algunas aplicaciones de gráficos computacionales en 3D, como por ejemplo para 3ds Max.

ANEXOS

A	Anexo 1. Metodología BIM	77
B	Anexo 2. Diseño del edificio en 3ds Max.....	89
C	Anexo 3. Lenguaje de programación. Códigos en C Sharp	99

Anexo 1. Metodología BIM

Título: Metodología BIM

Autor: Soraya Jeannette Araujo Criollo

Fecha: 27 de febrero 2017

Lugar: Barcelona

Glosario

UNIX: Es uno de los sistemas operativos más importantes en la historia de la computación. Se define como un sistema operativo de propósito general, multiusuario, e interactivo para los ordenadores PDP-11/40 y 11/45 de la Corporación Digital Equipment, es decir, es un sistema operativo portable, multitarea y multiusuario
--

XENIX: Se trata de un sistema operativo antiguo creado por Microsoft XENIX, fue diseñado para ser ejecutado sobre microordenadores

CAD: Proviene del nombre de Computer-Aided Design. Se define como el uso de programas informáticos para crear representaciones gráficas de los objetos físicos en dos o tres dimensiones. El software CAD puede ser especializado para aplicaciones específicas.

Palabras clave

3D, optimización, organización, productividad, fases, Revit, LOD, AR, VR, BIM conceptual, BIM construable.

Resumen

El Modelado de la Información para la Construcción o BIM es el método de trabajo que alinea de forma coordinada todos los factores que intervienen en un proyecto constructivo. La aplicación de este método conlleva a la comprensión del edificio en cada una de sus fases desde la fase de diseño, pasando por la ejecución hasta la de mantenimiento, y la participación de los diversos factores que actúen en la construcción, aportando información sobre el modelado y mantenimiento del edificio. La gran ventaja con la que juega BIM en un Edificio Inteligente, se da en su fase de diseño, ya que los distintos softwares que dan soporte al proyecto pueden predecir con gran acierto el comportamiento que tendrá el edificio ante determinados acontecimientos, siendo de vital importancia los componentes BIM que se añadan al proyecto.

Por ultimo hay que tener en cuenta también que el concepto BIM que se manejaba hace unos años ha ido evolucionado con el tiempo. Actualmente se podría decir que se maneja una estructura en torno a tres pilares que son las personas, los proceso y las tecnologías. Dentro de cada uno de estos grupos hay unas ideas que son claras e importantes a la vez y son; *la colaboración, intercambio de información y contrato (LOD) y la digitalización* donde se manejan las herramientas en cuanto a la creación de construcciones, análisis y gestión de la información, que lo maneja el Facility Management y Project Management.

Por otro lado, si miramos la evolución que ha tenido BIM en la Unión Europea, vamos a poder diferenciar tres ideas importantes que son; *innovación (Horizon 2020), uso e implantación [1]ón (Smart cities)*, que va asociado a la quinta dimensión que posee BIM.

1. Capítulo 1: Planteamiento

1.1. Introducción

BIM nace con el objetivo de digitalizar a través de elementos inteligentes, toda la información que sea necesaria para la realización de cualquier obra, como son los muros, pilares, ventanas, puertas, escaleras, etc., ya sea por ejemplo para un edificio de oficinas o un campo de fútbol. Convirtiéndose así en un prototipo digital de los elementos físicos del edificio. Esto nos va a poder permitir simular y entender el comportamiento de este en un entorno virtual antes de su construcción real.

Aunque dependiendo del autor o de la página donde se ha realizado la consulta, se va a poder ver varios puntos de vista diferentes acerca de cómo definen BIM, las más comunes son: *Se define como un software de visualización en tres dimensiones, lo describen también como un modelo 3D virtual de edificios, otros dicen que BIM es solo un proceso y por último, no es más que una colección de datos de un edificio organizados en una base de datos estructural que se puede consultar fácilmente de forma visual o numérica.* [1]

Uno de los grandes objetivos que se pretende alcanzar con este tipo de software es disminuir la pérdida de tiempo y los recursos en el diseño y la construcción.

1.2. Definición

BIM viene del acrónimo inglés “*Building Information Modeling*”, es un proceso de generación y gestión de datos durante todo su ciclo de vida, utilizando softwares dinámicos como revit, que crean modelados en tres dimensiones y a tiempo real, optimizando así el tiempo y los recursos destinados en las fases de diseño y construcción, con una notable mejora en la gestión de explotación de las infraestructuras.

La empresa húngara Graphisoft fue la pionera en la aplicación del concepto “BIM” con el programa ArchiCAD implementado bajo el nombre de Virtual Building, este software era capaz de crear dibujos tanto en 2D como en 3D. En 1978 se presentó la primera versión de Sigma Graphics, en el año de 1984 se creó un entorno completamente dedicado a la arquitectura y la construcción. Este software originalmente fue desarrollado para entornos multitarea tales como UNIX/XENIX y actualmente trabaja bajo sistemas operativos de Windows.

Si hacemos una comparación con la programación en CAD, se puede ver que estos utilizan sólo geometría en 2D o 3D sin diferenciar los elementos, sin embargo, BIM utiliza bibliotecas, estas son capaces de interpretar la interacción lógica entre los diferentes tipos de objetos y almacena la información referente a estos. Por tanto, este proceso abarca la geometría del edificio, las relaciones espaciales, la información geográfica, así como las cantidades y las propiedades de sus componentes. [2] Uno de los usos que se puede obtener con BIM es la ilustración de un proceso completo en edificación, del mantenimiento e incluso de la demolición (ya que ahora se reciclan más materiales). Además, en ámbitos laborales, detalles de componentes y secuencias de actividades de construcción pueden ser aislados y definidos donde la representación asistida por ordenador basada en objetos es un cambio sustancial en la tradicional elaboración basada en la representación vectorial.

Hoy en día este software es ofrecido por diferentes proveedores tecnológicos como: Tekla, Nemetschek, Sigma Design, Autodesk, StruCad de AceCad Software, Bentley Systems, Graphisoft, ACCA software, sds/2, por Design Data (líder en ingeniería de detalle), CAD Details, Dlubal Software, entre otros [2]

BIM marca un antes y un después en la nueva era que vivimos, ya que los profesionales de la Arquitectura, Ingeniería y Construcción “AEC” (Architecture, Engineering and Construction), van a poder no sólo ahorrar tiempo al crear y modificar sus proyectos, sino que también facilitará la

interacción al más alto nivel con sus colaboradores, asociados o colegas, al compartir contenidos específicos de cada especialidad en el mismo modelo BIM. [3]

1.3. Características



Figura 1. Ciclo de vida [4]

2. Capítulo 2: Fases de ciclo de vida de un proyecto en BIM

Un proyecto pasa por múltiples fases del ciclo de vida (PLPs¹). Dentro de cada fase se tendrá en cuenta una sub-fase donde las tres fases más importantes son. [5]

2.1. Fase de diseño

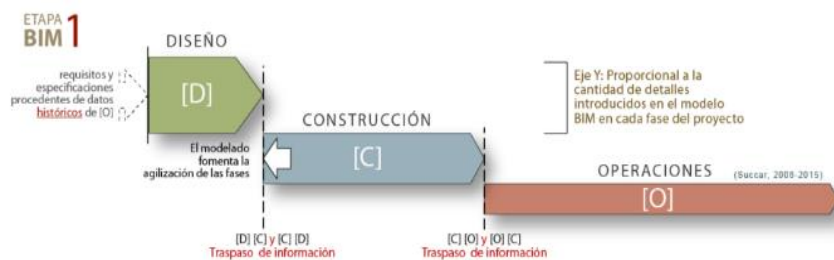


Figura 2. Fases del ciclo de vida del Proyecto en Etapa BIM 1 – proceso lineal

- Conceptualización, planificación y estimación de costes
- Diseño arquitectónico, estructural y de instalaciones
- Análisis, definición de detalle, coordinación y especificaciones.

2.2. Fase de construcción

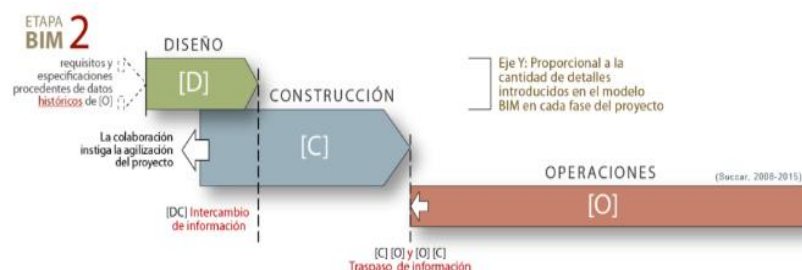


Figura 3. Fases del ciclo de vida del Proyecto en Etapa BIM 2 – modelo lineal

- Planificación y planos de taller

¹ Project Lifecycle Phases

- Construcción, fabricación, compras y aprovisionamientos
- Puesta en marcha, as-built y entrega.

2.3. Fase de operaciones

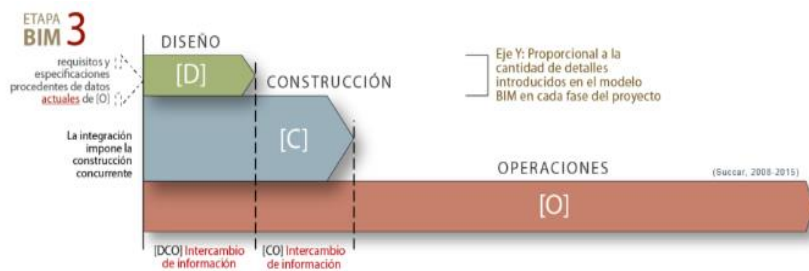


Figura 4. Fases del ciclo de vida del Proyecto en Etapa BIM 3 – modelo lineal

- Ocupación y operación
- Gestión de activos y mantenimiento de la instalación
- Desmantelamiento y reprogramación integral.

A pesar de categorizar las fases en tres grandes grupos una forma más sencilla de entenderlo y visualizarlo de una manera más real, se podría decir que también precisa, es dividiéndola en dos bloques

- Fase Virtual del Edificio Inteligente. Se identifica con la fase de diseño. Es el momento clave para conocer el verdadero alcance que tendrá el edificio y el grado de integración de la tecnología que se aplicará.
- Fase Real del Edificio Inteligente. Se identifica con la fase de ejecución y mantenimiento. Representa una puesta en escena del concepto del edificio. Tras la ejecución y culminación del proyecto, serán los administradores o propietarios quienes se encargarán del mantenimiento

Debe tener en cuenta que la unión fases-BIM a lo largo del proyecto es importante ya que, de otro modo, la probabilidad de equivocación en la fase de ejecución es alta. Como consecuencia, se tendría que reinvertir y contratar a un equipo técnico que junte toda la información, estudie el edificio y aporte aquellos datos que nos hubiera aportado un proyecto elaborado desde su planteamiento inicial con BIM.

2.4. Ventajas BIM vs método tradicional.

- Detecta previamente posibles conflictos en las distintas fases del proceso
- Aumenta y optimiza el tiempo
- Estima las repercusiones de cada solución en las distintas áreas, sin improvisar nuevos modelos
- Permite proyectar el calendario del proyecto y realizar un seguimiento durante la construcción. Facilita el cálculo de la eficiencia energética del edificio y sus instalaciones

3. Capítulo 3: Entorno BIM

3.1. Clasificación de dimensionamiento

Es la esencia de BIM, ya que marca el alcance de cada dimensión de forma acotada. Llama la atención que, aunque se ha consolidado como dimensión se podría hablar de ellas como fases, procesos del proyecto o simplemente como un producto. La figura 5 muestra tales dimensiones, donde la figura Project Management² va asociada desde 3D al 6D, y ya en el 7D la figura del Facility Management³ que es la figura más clara en una sola dimensión. [6]

3D	4D	5D	6D	7D
<i>Modeling</i>	<i>Scheduling</i>	<i>Estimating</i>	<i>Sustainability</i>	<i>Facility management</i>
Existing condition models	Project phasing simulation and installation	Quantity extraction to support cost estimates	Conceptual energy analysis	Life cycle BIM
Safety and logistics model	Visual validation for payment approval	Trade verification from fabrication models: Structural steel – Rebar – MEP	Detailed energy analysis	BIM As-built
Animation and rendering		Value engineering	Sustainable element tracking	BIM embedded
BIM driven prefabrication		Prefabrication solution	LEED tracking	BIM maintenance
Laser accurate BIM driven field layouts				Plans and technical support

Figura 5. BIM Dimensions

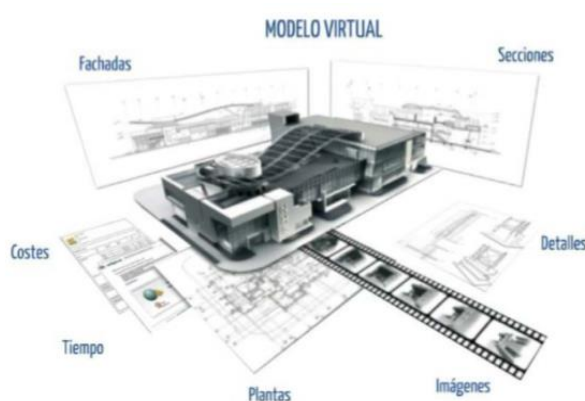


Figura 6. NHS Building — paastudio, CA, USA

²También es conocido como director del proyecto. Es la persona que tiene la responsabilidad total del planeamiento y la ejecución acertados de cualquier proyecto [9]

³ Es el profesional que se encarga de diseñar o adaptar los espacios e infraestructuras de una empresa para que los empleados desempeñen sus actividades con la mayor eficacia posible. [11]

3.2. Evolución de cambio de CAD a BIM

BIM y CAD representan dos aproximaciones diferentes al diseño arquitectónico y su documentación.

Las aplicaciones de **CAD (Computer Aided Design)** imitan el proceso de "papel y lápiz", son creados de forma independiente por lo que cada cambio en el diseño debe revisarse y modificarse manualmente en cada uno de ellos. Las aplicaciones **BIM (Building Information Modeling)** imitan el proceso real de construcción.

En lugar de crear dibujos en 2D se construyen los edificios de forma virtual modelándolos con elementos reales de construcción, como muros, ventanas, forjados, cubiertas, etc. Esto permite a los arquitectos diseñar edificios de la misma forma en que son construidos ya que como todos los datos están guardados en el modelo virtual central, los cambios en el diseño son automáticos consiguiendo así el incremento de la productividad y mejorando la coordinación. [7]

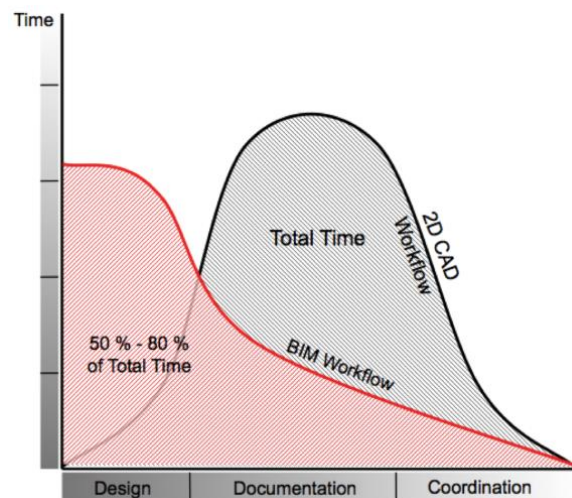


Figura 7. Optimización BIM vs CAD

4. Capítulo 4: Requisitos de selección más adecuados

Una de las ideas que hay que tener en cuenta es que de todas las soluciones que actualmente hay en el mercado, muchas de ellas no cumplen con los requisitos para que sea una solución fiable y verdadera. [7]

4.1. Profundidad

Cualquier modelo BIM, tiene que tener la capacidad de cubrir todas las necesidades a lo largo de su ciclo de vida. Hay varios niveles y profundidades para crear modelos de edificios en 3D, empezando por los modelos de comunicación hasta los modelos de información inteligente "reales" de la construcción.

La diferencia radica en que los modelos creados para la visualización no contienen más que la geometría 3D y las descripciones de los materiales necesarios. Los modelos reales de BIM, además de la geometría, también contienen una gran cantidad de información adicional necesaria para coordinar, documentar, listar y gestionar el edificio basándose en el modelo BIM inteligente.

4.2. Flujo de trabajo completo

Debe cubrir completamente el flujo de trabajo BIM sin tener que estar cambiando de herramienta y/o de flujos de trabajo en mitad del proyecto. Por lo tanto, es una cuestión vital tanto para el diseño, la documentación y para la realización y el funcionamiento del edificio cuando se trabaja bajo un entorno de trabajo basados en BIM.

4.3. Interoperabilidad

Los archivos que se generen deben ser compatibles con el resto de archivos que se generan en el flujo de trabajo. Es decir, es especialmente necesario en el caso de equipos de diseño interdisciplinarios que necesitan colaborar en diferentes aspectos del diseño del mismo edificio. Al seleccionar su herramienta BIM es crítico saber de qué manera su herramienta BIM soporta los estándares abiertos y flujos de trabajo abiertos que permiten la coordinación con los consultores, sin tener en cuenta el tipo o versión de su aplicación de diseño (herramienta BIM).

4

4.4. Colaboración basada en el modelo a tiempo real

El diseño constructivo es un proceso de equipo en la mayoría de los casos. Por lo tanto, compartir el diseño en BIM requiere de un enfoque totalmente nuevo que involucra a servidores BIM activos que ofrecen un acceso a tiempo real de forma paralela para todo el equipo de trabajo. Donde hay que tener en cuenta, la capacidad que tiene este, para soportar el trabajo de diseño ya que tiene que ser compartido a tiempo real.⁵

4.5. Rendimiento

La forma en que una solución BIM puede soportar este crecimiento depende de la habilidad para aprovechar las últimas mejoras en los hardware y en cómo manejar el modelo, se tiene que tener en cuenta el tamaño máximo soportado o recomendado para el proyecto.

4.6. Estándares locales y contenido local suficiente

Los proyectos de edificación no son piezas de diseño únicas, pero la forma en que son diseñadas varía según el país. Cuando se seleccione la herramienta BIM se debe interesar también por saber y comprobar de qué forma la solución BIM soporta los estándares de diseño locales y si tiene contenidos locales como elementos constructivos inteligentes, plantillas de listados y esquemas locales y grupos de atributos que siguen los estándares locales.

5. Capítulo 5: Niveles de desarrollo según el marco europeo

Es un documento que categoriza la precisión y contenido de un modelo BIM, de tal forma que los agentes sean conscientes en cada momento del uso y de las limitaciones de dicho modelo. No se trata de un sustituto del Plan de ejecución BIM (BEP⁶), sino de una herramienta de ayuda para categorizar el modelo BIM en cada fase.

Aunque los niveles que se marcan en LOD Specification son seis, en España se ha categorizado en 5 grupos: [8]

⁴ Para saber más sobre los flujos de trabajo “open BIM” visite www.graphisoft.com/openbim.

⁵ Sistema de compartición del Servidor BIM de GRAPHISOFT BIM Server visite www.graphisoft.es/bim_server/

⁶ BIM Execution Plan

Level of Development				
LOD 100	LOD 200	LOD 300	LOD 400	LOD 500
<i>Anteproyecto</i>	<i>Proyecto básico</i>	<i>Proyecto previo a la ejecución</i>	<i>Proyecto de ejecución</i>	<i>Planos As - built</i>
Es un diseño conceptual, el modelo aportara una visión general, básicamente aportara el volumen, la orientación y área	Aportará una visión general con información de magnitudes aproximadas, tamaño, forma, localización y orientación. El uso que se da es simplemente aumentar la capacidad de análisis	El elemento de modelo se representará gráficamente en el modelo como un sistema específico, objeto o ensamblaje en términos de cantidad, tamaño, forma, ubicación y orientación.	Contiene el detalle necesario para la fabricación o construcción. Donde el nivel de decisiones es exacto.	Este último nivel de desarrollo representa el proyecto, ya que se ha construido, con las condiciones conforme a obra. El modelo es adecuado para el mantenimiento y funcionamientos de la instalación.

Figura 8. Niveles de Level Of Developed en España/ ARUP España [4]

Hay que tener en cuenta que la propia LOD Specification advierte de la no correspondencia entre LOD y fases de proyecto, además de indicar que los LOD son aplicables a elementos y no deberían ser usados para catalogar la totalidad del modelo.

6. Capítulo 6: Relación del entorno AR/VR Y BIM

6.1. Realidad virtual – BIM

La realidad virtual es un entorno de escenas u objetos de apariencia real, generado mediante tecnología informática, que crea en el usuario la sensación de estar inmerso en él, ofrece en cada momento la posibilidad de crear una experiencia de inmersión personal única, a través de un recorrido virtual fotorrealista. La implementación de BIM a RV, pasa por gestionar de forma eficiente los activos, manteniendo así el control absoluto de los inmuebles y sus costes asociados, aumentando así el rendimiento y la productividad. Por tanto, contar con un modelo BIM – AR es vital para que el proyecto esté integrado en un potente entorno virtual 3D. [9]

El tándem de VR y BIM, se convierte en una potente herramienta de venta a la hora de comunicar un proyecto al cliente. Contar con un modelo 3D basado en BIM permite controlar una serie de datos técnicos que al implantarse con VR permite vivir una experiencia dinámica y real, modificando acabados, puntos de luz, distribución, medidas de muebles, entre otros. [1]

6.2. Aplicaciones de la VR en el sector de la construcción BIM.

- Herramientas de marketing y relación a través de la exploración de escenas virtuales construidas o por construir
- Reproducción de lugares históricos y de patrimonio
- Recreación o visualización de proyectos de paisajismo
- Configuración de elementos de interiorismo
- Estudio del comportamiento estructural de edificaciones sometidas a esfuerzos de distinta índole
- Mantenimiento y control del edificio

6.3. Realidad aumentada – Entorno BIM

Al contrario que sucede con la realidad virtual, la realidad aumentada, nos va a proporcionar información a tiempo real de actividades reales que se están llevando en cada momento, si esta herramienta la implementamos con un software de modelización, se convierte en una herramienta muy útil tanto para arquitectos como ingenieros, ya que por ejemplo una vez modelado un edificio y colocado todos sus componentes inteligentes cuando, se realice una visita la persona física a través de su dispositivo móvil o a través de unas gafas como por ejemplo las google glass, van a poder interactuar y ver entre otras cosas el avance de la obra, maquinaria utilizada, que trabajos se están realizando en cada momento, o por ultimo si hay alguna situación de riesgo..

Otro claro ejemplo de la combinación de ambas se puede ver en la figura del Facility Management ya que este podría introducir la vida útil de las lámparas o visualizar las instalaciones. En el sector de la construcción se puede visualizar todos los elementos constructivos que forman parte de la obra, como su situación planos de montaje, posibles incidencias y como subsanarlas a tiempo real o facilitar la ubicación de recursos y planificación de obra y control de costes. [10]

7. Capítulo 7: Formato universal IFC

Industry Foundation Classes, es un formato de datos de especificación abierta. Nace con el propósito de convertirse en un estándar que facilite la interoperabilidad entre programas del sector de la construcción

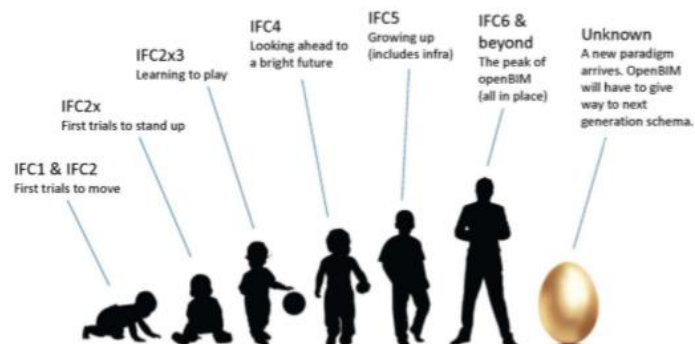


Figura 9. IFC- Levels of Maturity [3]

8. Conclusión

Una de las ideas más importantes que se debe tener en cuenta desde el primer momento que se va a trabajar con Building Information Modeling es el nivel de madurez que aumenta a partir de acciones sobre las personas, procesos y tecnología. España aún está a la cola en este tipo de modelado ya que este concepto aún es novedoso y no muy conocido. A través de los distintos work shop que se están realizando en varias ciudades españolas, se está dando a conocer poco a poco, las ventajas que tiene implementar este modelado al trabajo diario. Además, en el último año, se está contando también con el apoyo del gobierno español, aunque hay que decir, que de momento no se ha tendido la mano a alguna ayuda por parte del gobierno a las empresas. En otros países como por ejemplo Finlandia, Reino Unido o Incluso Estados Unidos, empezaron antes con la implementación de BIM y en la actualidad lo ven como algo natural. Incluso el gobierno finlandés ha invertido una parte de su presupuesto a ayudar a las empresas para que implanten BIM ya que con recursos se consigue aumentar el nivel de madurez BIM a mayor velocidad.

Por otra parte, la Unión Europea está trabajando para reducir el GAP BIM ⁷entre países, para ello ha creado este año un UE BIM TASK GROUP, que se encarga de ayudar a la administración para poder así eliminar las divergencias e igualar o equiparar los niveles a los que se maneja en la Unión Europea. Otro punto importante es el sentido común, necesario para saber para qué queremos BIM y no implementarlo simplemente por hacerlo.

Para terminar, diré que BIM es el futuro tanto en el ámbito de la ingeniería como de la arquitectura, ya que es capaz de gestionar de manera eficaz, la realidad compleja del sector de la construcción. Sin embargo, aún se tiene que seguir mejorando ya que su coordinación aun es lenta. En definitiva, hay que fomentar más la formación de técnicos en BIM, como ya se está haciendo en otros países.

Referencias

- [1] 24 Studio - Realidad Virtual – BIM , el gran cambio inmobiliario, «24studio in BIM, Consultoría, Implementación, VIZ, VR,» 13 Mayo 2016. [En línea]. Available: <http://www.24studiolab.com/realidad-virtual-bim/>. [Último acceso: 21 Junio 2016].
- [2] Building Information Modeling, «Wikipedia,» 20 Junio 2016. [En línea]. Available: https://es.wikipedia.org/wiki/Modelado_de_informaci%C3%B3n_de_construcci%C3%B3n.
- [3] 4M BIM, «qsai BIM España,» [En línea]. Available: <http://www.es.4mbim.com/what-is-bim/>. [Último acceso: 20 Junio 2016].
- [4] B. Community, Interviewee, *El caso de 'BIM para Terminales de Graneles Sólidos', de ARUP España*. [Entrevista]. 28 Abril 2016.
- [5] E. 1. E. d. B. e. l. f. d. c. d. v. d. u. proyecto, «Espacio Lean BIM. Construcción colaborativa,» Espacio Lean BIM, 27 Abril 2016. [En línea]. Available: <http://www.espacioleanbim.com/episodio-10-efecto-del-bim-las-fases-del-ciclo-vida-proyecto/>. [Último acceso: 21 Junio 2016].
- [6] Las dimensiones del BIM, «Antonio Flores Arquitectura,» [En línea]. Available: <https://antoniofloresarquitectura.wordpress.com/2015/07/15/las-dimensiones-del-bim/>. [Último acceso: 21 Junio 2016].
- [7] Graphisoft Open BIM, «Graphisoft,» [En línea]. Available: https://www.graphisoft.es/archicad/open_bim/about_bim/. [Último acceso: 20 Junio 2016].
- [8] Breve introducción a los niveles LOD en modelos BIM , «Sánchez Matamoros Arquitectos,» 14 January 2015. [En línea]. Available: <http://www.sanchez-matamoros.com/blog/2015/1/14/breve-introduccion-a-los-niveles-lod-en-modelos-bim>. [Último acceso: 21 Junio 2016].
- [9] Project Management. Gestión de proyectos , «Enrique López Salom. Arquitecto Técnico,» [En línea]. Available: <http://elopez->

⁷ Mal desempeño de BIM, cuando se trabaja en grandes proyectos.

projectmanagement.blogspot.com.es/2012/01/definiciones.html. [Último acceso: 21 Junio 2016].

- [10] P. J. Abós, «Realidad aumentada en arquitectura mediante modelado 3D,» 23 Mayo 2014. [En línea]. Available: <http://pilarjimenezabos.com/realidad-aumentada-en-arquitectura/>. [Último acceso: 21 Junio 2016].
- [11] Barcelona activa. Facility Management , «Ajuntament de Barcelona,» [En línea]. Available: <http://w27.bcn.cat/porta22/es/fitxes/F/fitxa5552/facility-manager.do>. [Último acceso: 21 Junio 2016].
- [12] B. d. g. B. d. u. d. Aragón, «Realidad Virtual y su aplicación al BIM,» 11 Mayo 2016. [En línea]. Available: <http://www.grupobimaragon.es/realidad-virtual-y-su-aplicacion-al-bim>. [Último acceso: 21 Junio 2016].

Índice de figuras

Figura 1. Ciclo de vida [4]	79
Figura 2. Fases del ciclo de vida del Proyecto en Etapa BIM 1 – proceso lineal	79
Figura 3. Fases del ciclo de vida del Proyecto en Etapa BIM 2 – modelo lineal.....	79
Figura 4. Fases del ciclo de vida del Proyecto en Etapa BIM 3 – modelo lineal.....	80
Figura 5. BIM Dimensions.....	81
Figura 6. NHS Building — paastudio, CA, USA.....	81
Figura 7. Optimización BIM vs CAD	82
Figura 8. Niveles de Level Of Developed en España/ ARUP España [4]	84
Figura 9. IFC- Levels of Maturity [3]	85

Anexo 2. Diseño del edificio en 3ds Max

Título: Diseño del edificio en 3ds Max

Autor: Soraya Jeannette Araujo Criollo

Fecha: 27 de febrero 2017

Lugar: Barcelona

Resumen

A partir de la documentación teórica previa y primeros ensayos, el presente documento recoge el trabajo realizado sobre el edificio de estudio. El objetivo a conseguir en esta práctica es poder tener una visualización del esqueleto del edificio. Para ello lo primero que se procede a hacer es abrir el archivo de CAD del edificio, exportarlo a FBX o dejarlo en su formato de origen y a partir de ahí ya se puede empezar a trabajar con 3DS MAX.

Hay que destacar que, si se cambia el archivo a formato FBX, se va a poder trabajar tanto en Revit, Unity y 3DS MAX, entre otros programas. En este caso como se va a poder elegir, lo que se plantea es, cuál de estos tres programas es más eficiente para el trabajo a realizar. Cada uno de estos tres programas tiene sus ventajas e inconvenientes ya que por ejemplo Unity es muy eficiente para la visualización de objetos en 3D, pero por el contrario si queremos importar directamente la estructura, se puede tener un problema de pérdida de capas, por lo que parece recomendable trabajar primero el archivo con 3DS MAX y luego con Unity. Otra de las formas de trabajo sería con Revit, en teoría mucho más sencillo que en 3DS MAX, pero el hándicap de dejarlo todo como una poli-línea. Para una estructura un tanto compleja como es el caso que presento, no es conveniente ya que se invierte más tiempo del necesario y no se garantiza que todo sea una poli-línea. Por lo tanto, para finalizar a modo resumen y de acuerdo con los experimentos realizados la opción que se ha mostrado ha sido trabajar el archivo es con 3DS MAX.

Palabras clave

3D, layers, render.

1. Capítulo 1: Introducción

Este documento muestra la exploración de las posibilidades de la realidad aumentada para el sector de la construcción. En particular, uno de los primeros objetivos que se es disponer de una visualización a tiempo real por medio de programas de diseño y visualización en 3D para poder aplicarlos sobre cualquier edificio, a través de un tour virtual, y así poder evaluar insitu las posibles mejoras, daños o incluso simplemente revisar para poder combinar la visualización real con los elementos no visibles, tipo de instalaciones, vigas o en el cimiento, por ejemplo.

Para ello debemos plantearnos, el tipo de formato con el que trabajar, así como el programa de modelado y visualización de la edificación para finalmente, como realizar la comunicación entre programas, por ejemplo, de un CAD tipo 3DS MAX a Unity, lo que requeriría emplear el formato de intercambio en FBX para no perder propiedades en las capas.

2. Capítulo 2: Ejecución

Decidimos trabajar con 3DS MAX. Por defecto, aparece cuatro sub-ventanas diferentes, tres de ellas permiten tener una vista de las diferentes perspectivas y la otra una visual en 3D. [1] [2]

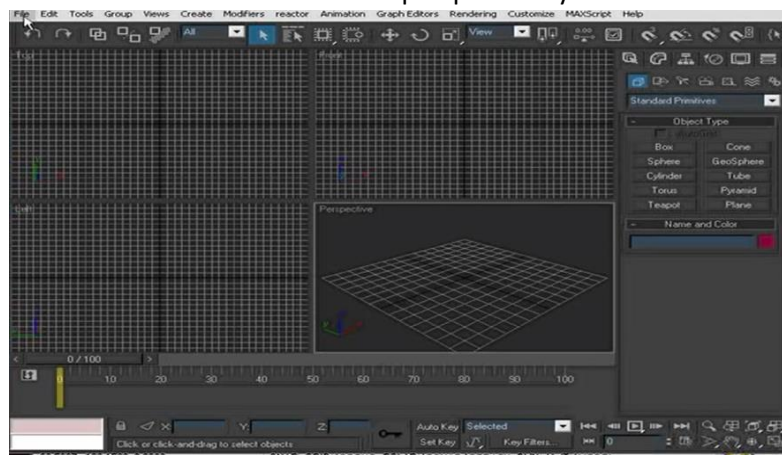


Figura 1. Visualización de la ventana principal

Para abrir el archivo se ha elegido dejarlo en el formato de origen, es decir, en DWG, para ello se ha de ir a file, importar archivo, elegimos el archivo que queremos y se procede a aceptar.

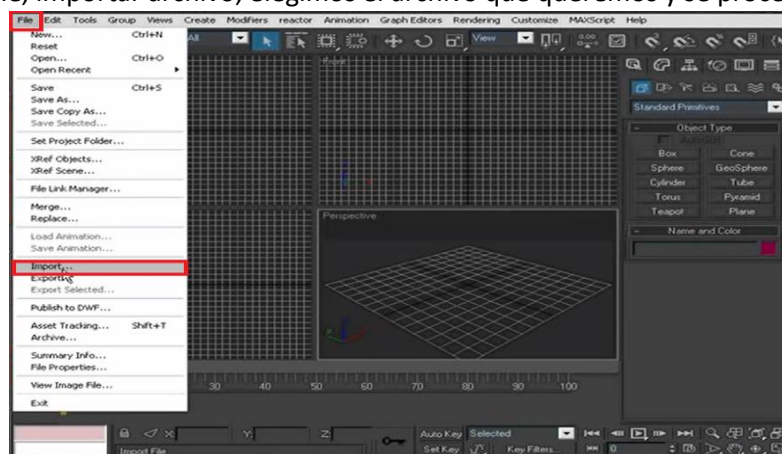


Figura 2. Importar CAD en formato FBX

Al finalizar se va abrir automáticamente otra pestaña que va referida a las propiedades que se quiere dar al archivo. Una de las cosas que se ha de tener en cuenta son las unidades y las capas.

Normalmente se va a trabajar en metros. Y en cuanto a las propiedades hay dos formas de abrirlo, por entidades y por capas. En mi caso se ha elegido abrir por capas ya que los objetos del CAD están definidos como tal y van a ser reconocidos por 3DS como un solo objeto, aunque sean varios. Es decir, los unifica.

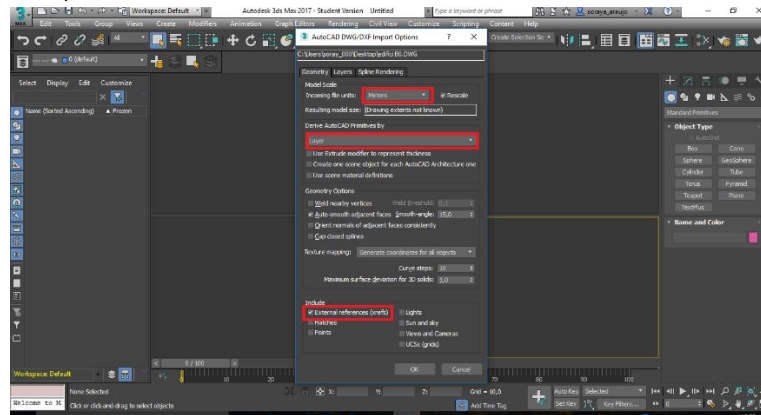


Figura 3. Propiedades del archivo

En un principio solo es recomendable cambiar las unidades y las capas del proyecto, todo lo demás se quedaría igual, ya que le archivo puede hacerse demasiado pesado y ralentizar el trabajo. Aunque se ha dicho que lo demás se deje igual, en este caso de estudio vamos también a marcar “external reference” ya que de lo contrario al Renderizar aparecería una ventana sin nada. Por tanto, la pestaña de geometría quedaría constituida de la siguiente manera.

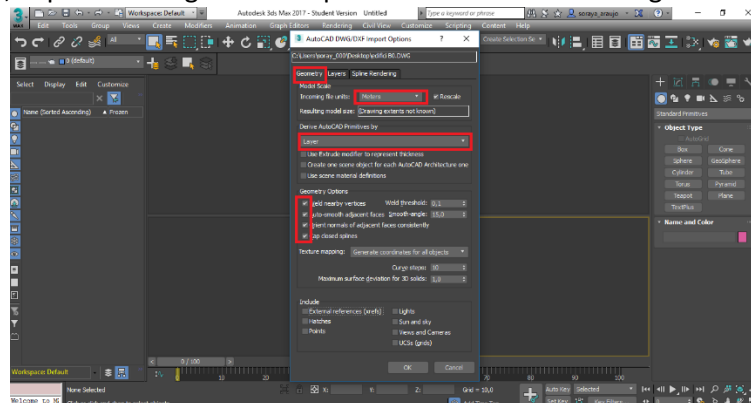


Figura 4. Visualización de la geometría

Una vez definida la geometría, la siguiente pestaña va referida a las capas, de manera predeterminada viene marcada en “skip all frozen layers”, esta hay que cambiarla.

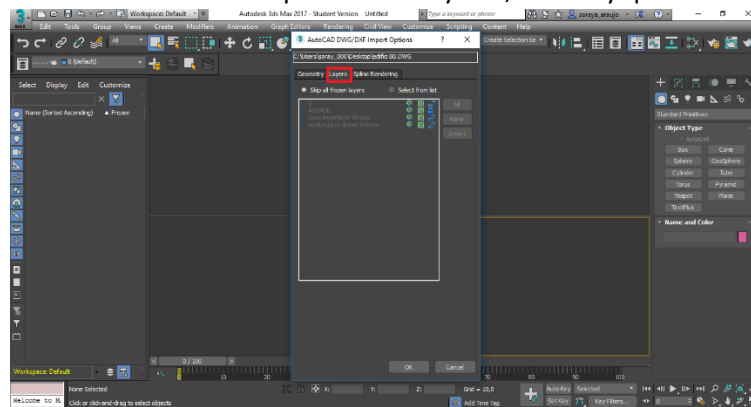


Figura 5. Visualización de las capas

En esta pestaña se va a poder elegir las capas que se quieran importar, en un principio como podemos ver en la figura anterior aparece por defecto “skip all frozen layers” y tal y como se ha dicho hay que cambiarla por “select from list” para así poder trabajar con más libertad

Es recomendable borrar las líneas guías de 0 ya que no sabemos si en el proyecto CAD se ha podido pasar por alto alguna línea guía, por tanto, aquí se podrá eliminar de manera total.

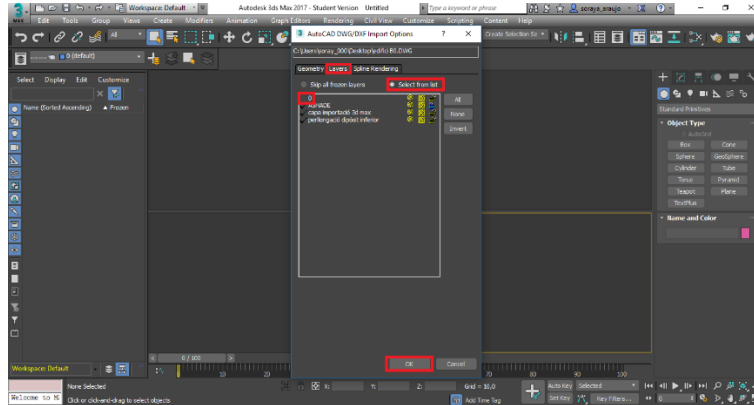


Figura 6. Parámetros de trabajo

La última de las pestañas, de momento no se va a modificar y se va a dejar tal cual, ya que modificarla o no, va a depender de lo que se busque en el proyecto.

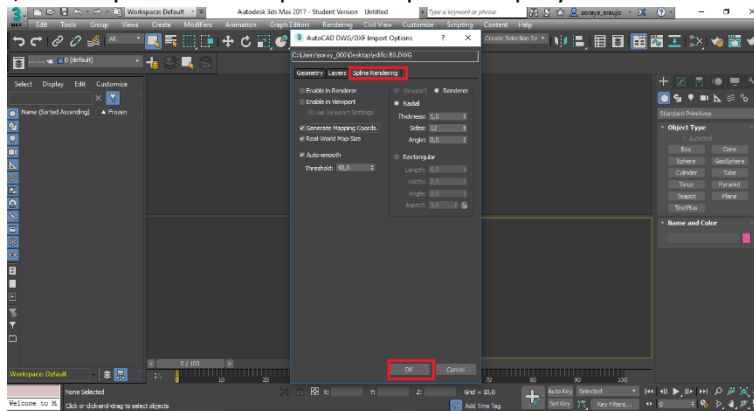


Figura 7. Propiedades de Spline Rendering

Una vez terminada la definición de todas las pestañas, se va a hacer doble clic en ok para que se lleve a cabo la importación.

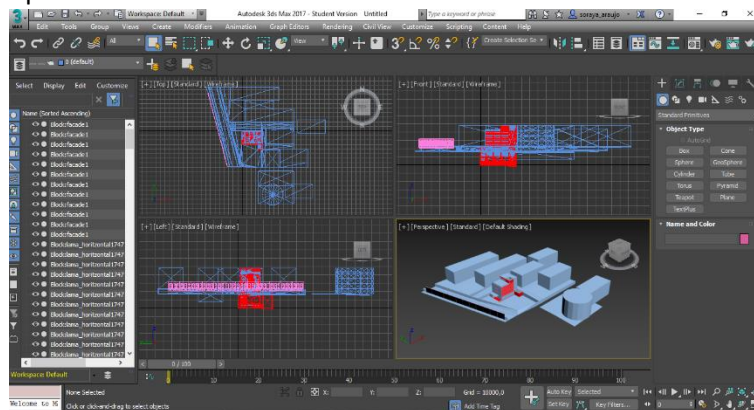


Figura 8. Visualización edificio CIMNE

Para comprobar si la importación se ha realizado de manera correcta, una de las formas para comprobarlo sería por medio de la renderización para ello se ha de ir a render y ejecutar. Como se ve en la figura siguiente al terminar el proceso de ejecución, aparece una pantalla en negro, por tanto, en un primer momento se pensaría que hay algo mal, pero, todo lo contrario, ya que solo depende de la visualización.

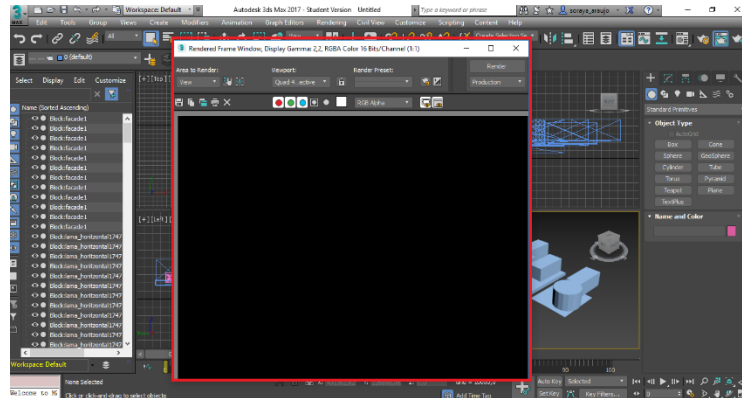


Figura 9. Rendelizado

Por tanto, lo se tendría que hacer es, ir a render > material editor > compact material editor > ok. En esta se va abrir una nueva pestaña nueva, donde habrá que darle unas propiedades determinarlas y posteriormente ejecutarlas.

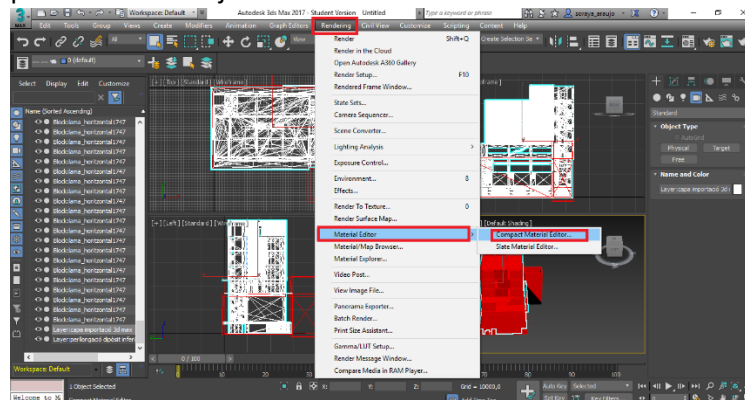


Figura 10. Editor de materiales

En esta pestaña vamos a poder asignar materiales para poder arreglar el pequeño inconveniente que se ha presentado. Para ellos habrá que hacer clic en estándar, seleccionamos Material/Map Browser > Materials > Standard > DoubleSided.

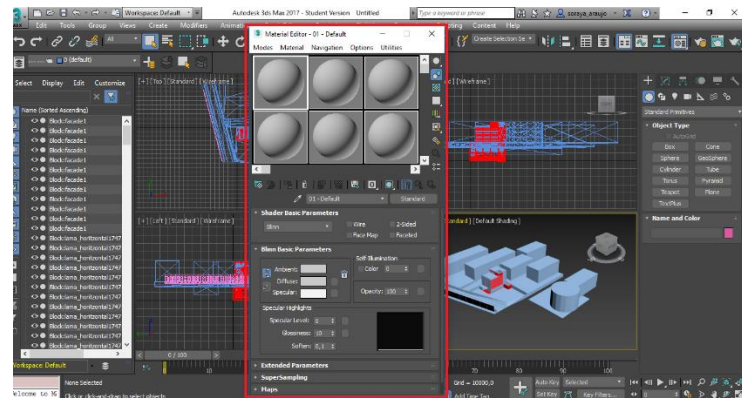


Figura 11. Designación de materiales

Ahora se va a seleccionar todo, eso se puede hacer en cualquiera de las cuatro vistas, en mi caso he elegido la cuarta vista, una vez este seleccionado todo, se va al botón de asignar materiales y vamos al primer material y automáticamente queda ya asignado, así sucesivamente se haría con el resto.

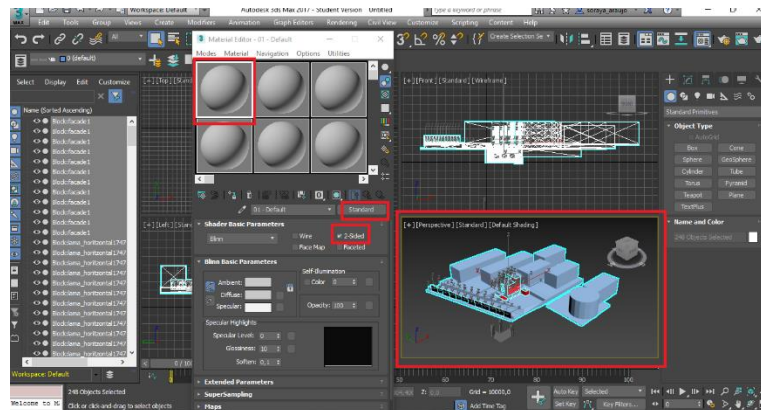


Figura 12. Propiedades de los materiales

Una vez asignado el material se va a ejecutar de nuevo Renderizar, hay que hacer doble clic en render ya que no se ejecuta de manera automática, como el primer caso. Como se puede ver todos los objetos están definidos ya de manera correcta, ya que se encuentran definidas en las dos caras.

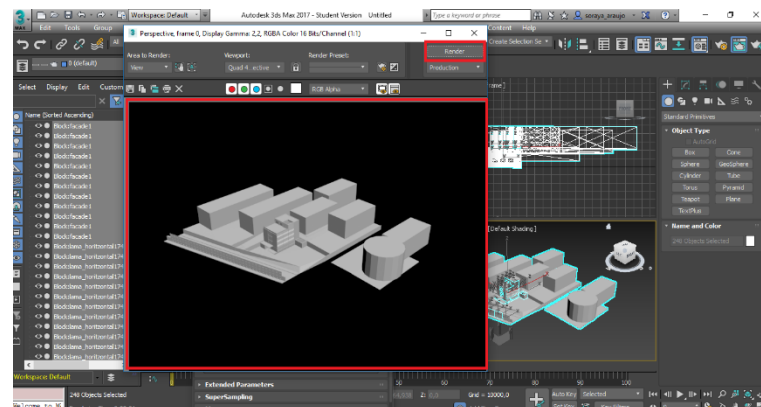


Figura 13. Visualización de Render

Una vez terminada la comprobación y visualización correcta del edificio, lo siguiente que se procede hacer es desactivar por un lado las capas, para poder ver el esqueleto del edificio y, por otro lado, el sólido hay que convertirlo en una estructura transparente para poder ver los elementos que componen su interior. [2]

2.1. Visualización de la estructura

Si tenemos el archivo en CAD, se puede ver de manera directa desde este, ya que una de las ventajas que tiene el trabajar con AutoCAD es precisamente la visualización sobre todo en 2D, pero también en 3D, si en el caso de no tener el plano en CAD, lo que tendríamos que hacer es recopilar toda la información que se pueda tener y a partir de ahí trabajar con los datos.

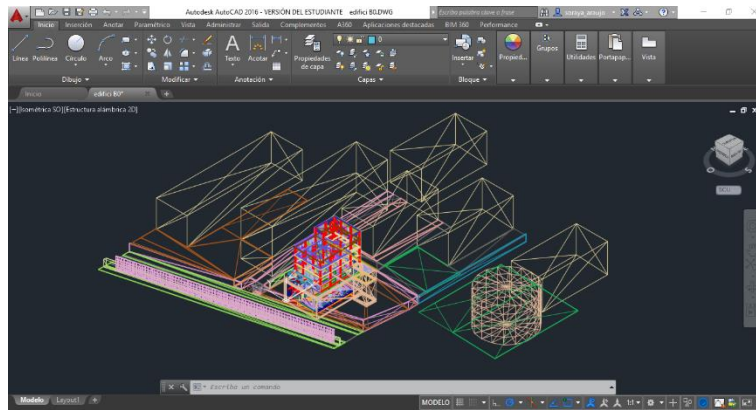


Figura 14. Esqueleto de la estructura

2.2. Visualización de la estructura de forma traslúcida

Como se ha dicho previamente hay dos puntos de vista desde los cuales se puede analizar, ambos se han de trabajar a partir de 3DS MAX.

El primero es muy sencillo, ya que una vez definida la estructura lo único que tenemos que hacer es seleccionar la estructura en cualquiera de las vistas y presionar a continuación ALT+X, quedando de la manera siguiente

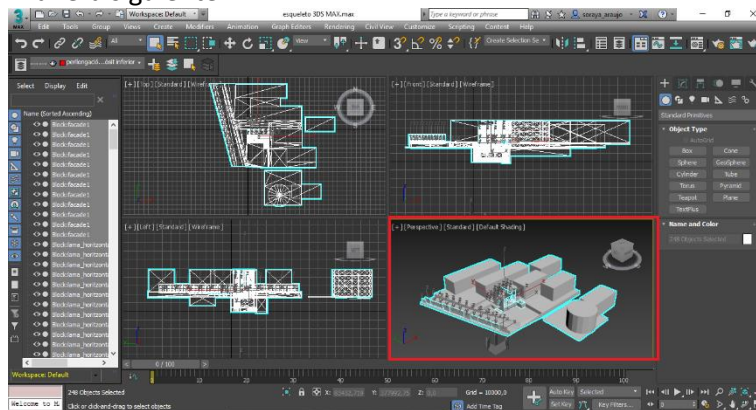


Figura 15. Vista seleccionada para la ejecución del comando

Una vez ejecutada, se va a poder ver el interior de la estructura y todos sus elementos y componentes que se encuentran en él, hay que destacar que al ejecutar este comando todas sus vistas van a cambiar de color, esto quiere decir que cambia en todas las perspectivas.

Por tanto, la visualización quedaría de la siguiente manera.

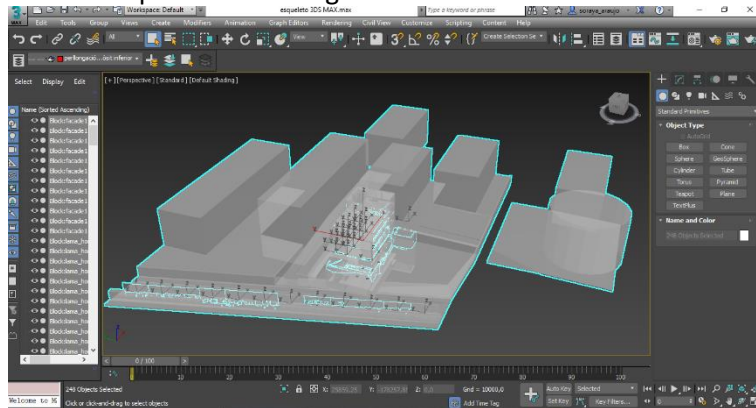


Figura 16. Visualización de la estructura traslúcida

Si se sigue haciendo zoom, se va a poder ver más en detalle, por ejemplo, sus pilares, los huecos de las escaleras, ventanas, paredes, puertas pasillos... etc.

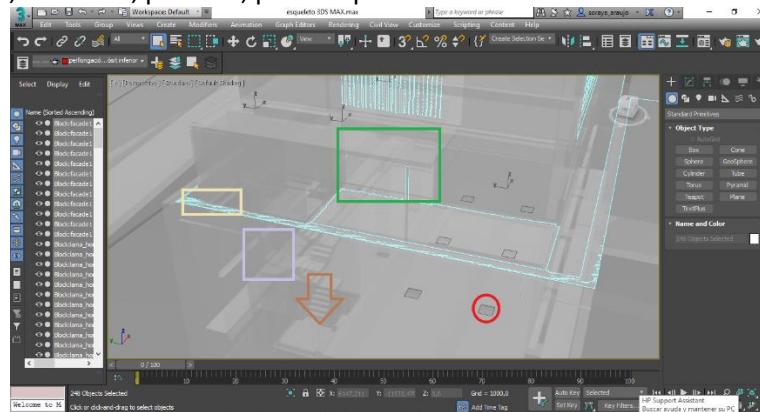


Figura 17. Zoom del detalle del edificio

La segunda forma es a partir del comando Slice, con este vamos a poder hacer una especie de corte en el edificio y ver su interior. Para usar el comando Slice se necesita tener todo como un objeto, por ejemplo, el techo sería un objeto, el suelo otro objeto y todo el perímetro del edificio otro objeto. Ahora vamos a attach, para encontrar este comando primero se ha de seleccionar el objeto a continuación vamos a grupo > assembly > attach y se va a aplicar tanto al suelo como al techo una vez seleccionado estos, se ejecuta y se puede comprobar que ya trabaja como un solo objeto.

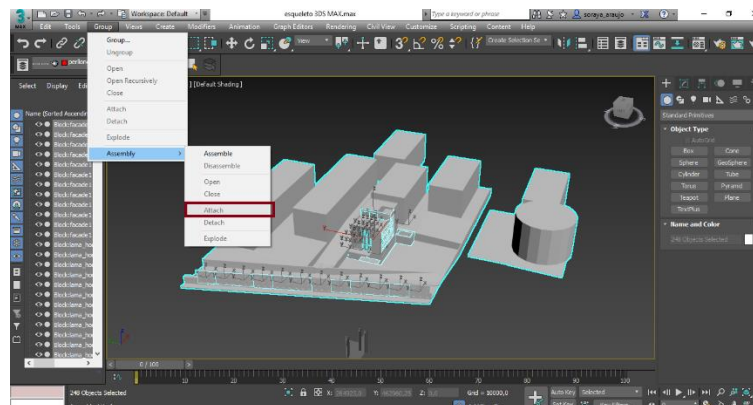


Figura 18. Ejecución de comando Slice

A pesar de las dificultades que nos hemos encontrado, he procedido a ejecutar Slice, para ello primero hay que seleccionar el objetivo, ir a modificar y se ejecuta Slice. Se puede ver de manera gráfica con la siguiente figura.

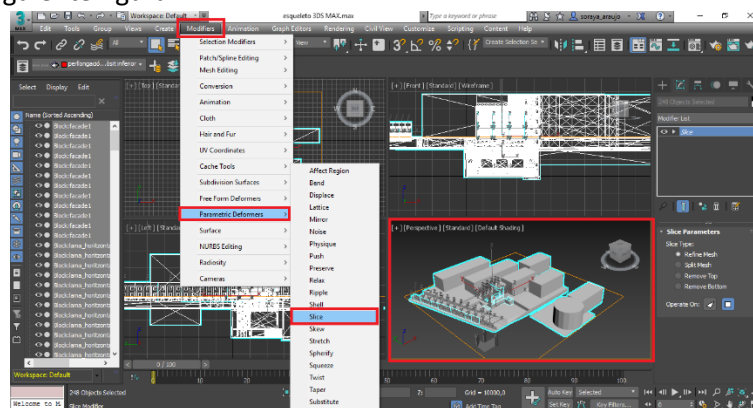


Figura 19. Ejecución de comando Slice

Como se puede ver en la figura 18, alrededor del objeto aparece un rectángulo, para poder trabajar con este, se ha de ir al panel que aparece al lado derecho ampliamos la pestaña Slice y también a continuación se selecciona Slice Plane, una vez ejecutado se puede observar que el color cambia a un tono amarillo

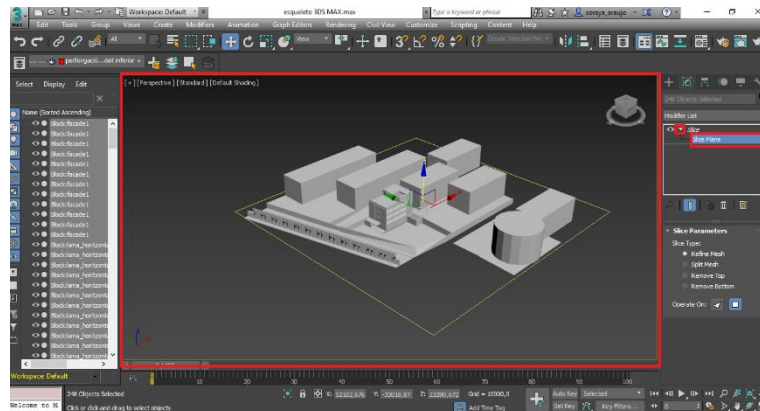


Figura 20. Ejecución de Slice Plane

A continuación, se va hace clic derecho en los ejes de coordenadas y se procede a rotar y a continuación si se va tiendo presionado angle snap toggle , se procede a rotar 90°

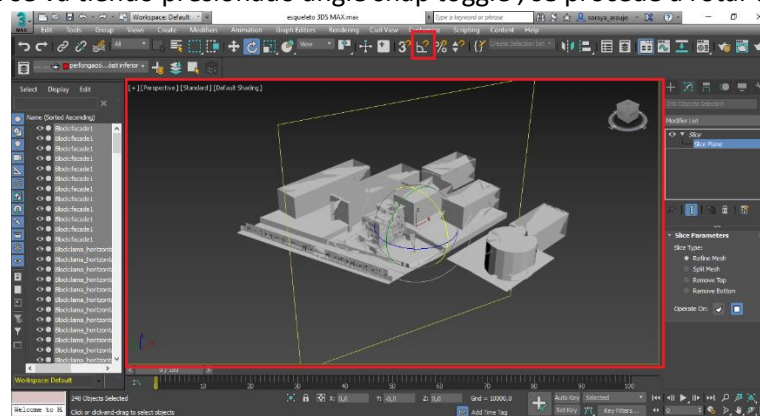


Figura 21. Rotación con un ángulo de 90°

Una vez rotado, se va a dar a la tecla w, para mover y desplazar o también se puede ejecutar este comando a través del icono de las cuatro flechas, al desplazar se puede observar aparece una línea nueva

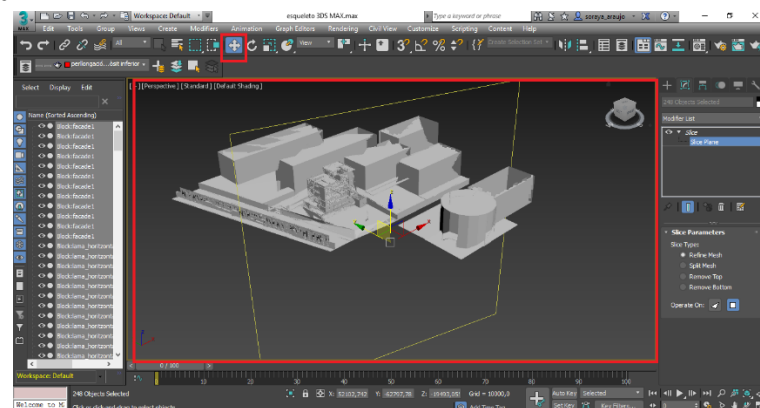


Figura 22. Desplazamiento del objeto

Con esta nueva línea, se va a proceder cortar el edificio, si se va al icono de remove bottom se va a borrar la parte de abajo del plano y si nos vamos moviendo a tiempo real va apareciendo la

edificación poco a poco, si se quiere ir un paso más allá se podría hacer una animación de cómo va cambiando.

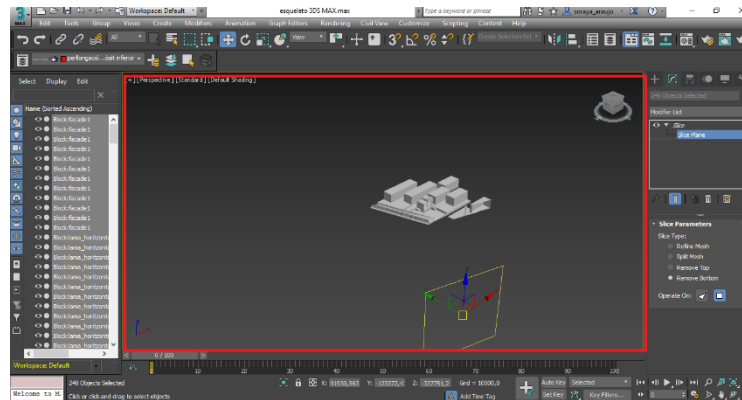


Figura 23. Desplazamiento a tiempo real de la estructura

Y así concluiría la práctica, ya que se podría ver el edificio por rebanadas y se podría tener una idea de lo que hay en el interior del edificio, hay que decir por último que podemos rotar con el ángulo que se quiera no es necesario que sea el de 90° , incluso se podría empezar a cortar desde arriba y no por el alzado.

3. Capítulo 3: Conclusiones

A lo largo de la realización de esta práctica se ha podido evaluar de una manera más específica el funcionamiento de 3DS MAX, una de las ideas primera ideas que se puede obtener de este estudio es el poder de visualización que se tiene, ya que, al ser un CAD de alta calidad, lo soporta de una manera bastante correcta, al contrario de lo que sucede con revit, ya que en este se pierde algunas de sus propiedades y previamente hay que trabajar también en CAD, cambiando todo a poli línea. Otra de las ideas principales que se puede obtener es que una vez ejecutado el comando Slice, se va a poder visualizar la estructura de manera sencilla ya que solo es jugar con los desplazamientos y los ángulos que se quiera dar, esto grosso modo es bueno ya que podemos ver cualquier parte del edificio que se quiera a tiempo real.

4. Capítulo 4: Referencias

- [1] Convertir objetos en translúcidos en 3D STUDIO , «Youtube,» [En línea]. Available: <https://www.youtube.com/watch?v=MPFiAXsOyJY>. [Último acceso: 27 Junio 2016].
- [2] Usando el Modificador Slice 3ds Max Tutorial, «Youtube,» 3dluistutorials , [En línea]. Available: https://www.youtube.com/watch?v=eAHF9tL_1QQ. [Último acceso: 27 Junio 2016].

Figuras

Figura 1. Visualización de la ventana principal	89
Figura 2. Importar CAD en formato FBX.....	89
Figura 3. Propiedades del archivo	90
Figura 4. Visualización de la geometría.....	90
Figura 5. Visualización de las capas.....	90
Figura 6. Parámetros de trabajo	91
Figura 7. Propiedades de Spline Rendering	91
Figura 8. Visualización edificio CIMNE	91
Figura 9. Rendelizado	92
Figura 10. Editor de materiales	92
Figura 11. Designación de materiales	92
Figura 12. Propiedades de los materiales	93
Figura 13. Visualización de Render	93
Figura 14. Esqueleto de la estructura	94
Figura 15. Vista seleccionada para la ejecución del comando	94
Figura 16. Visualización de la estructura traslúcida	94
Figura 17. Zoom del detalle del edificio	95
Figura 18. Ejecución de comando Slice	95
Figura 19. Ejecución de comando Slice	95
Figura 20. Ejecución de Slice Plane	96
Figura 21. Rotación con un ángulo de 90°	96
Figura 22. Desplazamiento del objeto	96
Figura 23. Desplazamiento a tiempo real de la estructura	97

Anexo 3. Lenguaje de programación. Códigos en C Sharp

Título: Lenguaje de programación. C Sharp

Autor: Soraya Jeannette Araujo Criollo

Fecha: 27 de febrero 2017

Lugar: Barcelona

1. ClickDeMouse

```
using UnityEngine;
using System.Collections;
public class ClickDeMouse : MonoBehaviour
{
    public string texto = "";
    public void Update()
    {
        OnMouseDown();
    }
    public void OnMouseDown()
    {
        if (Input.GetMouseButton(0))
        {
            texto = "Pilar";
        }
        else
        {
            texto = "";
        }
    }
}
```

2. ColisionesInstalacion

```
using UnityEngine;
using System.Collections;
public class ColisionesInstalacion : MonoBehaviour {
    public float m_Velocidad = 8F;
    public float m_VelocidadDeRotacion = 80F;
    public bool EnPiso = true;
    public int m_VelocidadDeSalto=200;
    public Rigidbody rb;
```

```
//private RaycastHit m_Golpe;
// private int count;
void Start()
{
    rb = GetComponent<Rigidbody>();
}
void OnTriggerEnter(Collider other)
{
    Destroy(other.gameObject);
}
void FixedUpdate()
{
    if (Input.GetButtonDown("Jump"))
        rb.velocity = new Vector3(0,4,0);
}
void Update()
{
    float translation = Input.GetAxis("Vertical") * m_Velocidad;
    float rotation = Input.GetAxis("Horizontal") * m_VelocidadDeRotacion;
    translation *= Time.deltaTime;
    rotation *= Time.deltaTime;
    transform.Translate(0, 0, translation);
    transform.Rotate(0, rotation, 0);
    RaycastHit m_Golpe;
    Vector3 physicsCentre = this.transform.position +
this.GetComponent<CapsuleCollider>().center;
    Debug.DrawRay(physicsCentre, Vector3.down * 1.0f, Color.green, 1);
    if (Physics.Raycast(physicsCentre, Vector3.down, out m_Golpe, 1.0f))
    {
        if (m_Golpe.transform.gameObject.tag != "Player")
        {
            EnPiso = true;
        }
    }
    else
    {
        EnPiso = false;
    }
    Debug.Log(EnPiso);

    if (Input.GetKeyDown(KeyCode.Space) && EnPiso)
```

```
    {  
  
this.GetComponent<Rigidbody>().AddForce(Vector3.up*m_VelocidadDeSalto);  
    }  
}  
  
// void OnTriggerEnter(Collider other)  
// {  
//     // if (Input.GetKeyDown(KeyCode.Space))  
//     {  
//         //this.GetComponent<Rigidbody>().AddForce(Vector3.up * 200);  
//     }  
// }  
}
```

3. ConfiguracionBoton

```
using UnityEngine;  
using System.Collections;  
public class ConfiguracionBoton : MonoBehaviour {  
    public void BotonJugar ()  
    {  
        Application.LoadLevel("INSTALACIONES");  
    }  
  
    public void BotonJugarAQR()  
    {  
        Application.LoadLevel("ARQUITECTURA");  
    }  
    public void BotonSalir ()  
    {  
        Application.Quit();  
    }  
}
```

4. Gráficos

```
using UnityEngine;
using System.Collections;
public class Graficos : MonoBehaviour
{
    public void Graficos1()
    {
        QualitySettings.currentLevel = QualityLevel.Fastest;
    }
    public void Graficos2()
    {
        QualitySettings.currentLevel = QualityLevel.Fast;
    }
    public void Graficos3()
    {
        QualitySettings.currentLevel = QualityLevel.Simple;
    }
    public void Graficos4()
    {
        QualitySettings.currentLevel = QualityLevel.Good;
    }
    public void Graficos5()
    {
        QualitySettings.currentLevel = QualityLevel.Beautiful;
    }
    public void Graficos6()
    {
        QualitySettings.currentLevel = QualityLevel.Fantastic;
    }
}
```

5. MensajePilares

```
#pragma strict

var mensaje : String; // mensaje q se va a mostrar
var entro : boolean = false;
function Start () {}
function Update () {}
function OnGUI()
{
    if(entro)
    {
        GUI.Label(Rect(Screen.width/2-100,50,200,30),mensaje);
    }
}

function OnTriggerEnter()
{
    entro = true;
}

function OnTriggerExit()
{
    entro = false;
}
```

6. PuertaDoble

```
using UnityEngine;
using System.Collections;
using System.ComponentModel;
public class PuertaDoble : MonoBehaviour {
    private float targesValue = 180;
    private float currentValue = 180;
    private float easing = 0.05f;
    public float AngleY = 270;
    public GameObject targes;

    void Update()
    {
        currentValue = currentValue + (targesValue - currentValue) * easing;
        targes.transform.rotation = Quaternion.identity;
        targes.transform.Rotate(0, currentValue, 0);
    }
}
```

```
    }  
    void OnTriggerEnter(Collider other)  
    {  
        targesValue = AngleY;  
        currentValue = 180;  
    }  
    void OnTriggerExit(Collider other)  
    {  
        targesValue = 180;  
        currentValue = AngleY;  
    }  
}  
// Puerta derecha
```

7. PuertaSimple

```
using UnityEngine;  
using System.Collections;  
public class PuertaSimple : MonoBehaviour {  
    private float targesValue=0;  
    private float currentValue = 0;  
    private float easing = 0.05f;  
    public float AngleY = 90;  
    public GameObject targes;  
    void Update()  
    {  
        currentValue = currentValue + (targesValue - currentValue) * easing;  
        targes.transform.rotation = Quaternion.identity;  
        targes.transform.Rotate(0, currentValue, 0);  
    }  
    void OnTriggerEnter(Collider other)  
    {  
        targesValue = AngleY;  
        currentValue = 0;  
    }  
    void OnTriggerExit(Collider other)  
    {  
        targesValue = 0;  
        currentValue = AngleY;  
    }  
}
```

```
    }  
}  
  
    // Puerta derecha
```

8. ScreenReveiver

```
using UnityEngine;  
using System.Collections;  
public class ScreenReciver : MonoBehaviour {  
    public string textScreen;  
    public TextMesh pantalla;  
    public string ClickDeMouse;  
    public void Update()  
    {  
        ClickDeMouse clickar = GetComponent<ClickDeMouse>();  
        textScreen = clickar.texto;  
        pantalla = (TextMesh)GameObject.Find("pilars planta  
baixa").GetComponent<TextMesh>();  
        pantalla.text = "" + textScreen;  
    }  
}
```

9. PlayerController

```
using UnityEngine;  
using UnityEngine.UI;  
using System.Collections;  
public class PlayerController : MonoBehaviour {  
    public float speed;  
    public Text countText;  
    public Text winText;  
    private Rigidbody rb;  
    private int count;  
    void Start()  
    {  
        rb = GetComponent<Rigidbody>();  
        count = 0;  
        SetCountText();  
        winText.text = "";  
    }  
    void FixedUpdate()  
    {  
        float moveHorizontal = Input.GetAxis("Horizontal");
```



```
float moveVertical = Input.GetAxis("Vertical");  
Vector3 movement = new Vector3(moveHorizontal, 0.0f, moveVertical);  
rb.AddForce(movement * speed);  
}  
void OnTriggerEnter(Collider other)  
{  
    if (other.gameObject.CompareTag("Pick-up"))  
    {
```

